Computational Linguistics and Talking Robots

Processing Content in Database Semantics

PREPRINT
Second Edition
corrected revised extended
Please cite as Preprint March 4, 2017

ROLAND HAUSSER

Friedrich Alexander Universität Erlangen Nürnberg

March 4, 2017

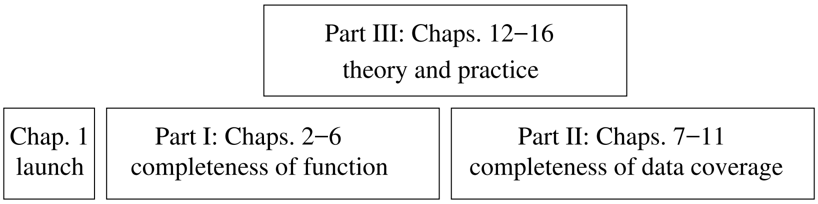
email: rrh@linguistik.uni-erlangen.de

coverphoto: Sphinx
by Jakob C. Schletterer (1699–1774)
Kloster Altenburg, 1735
Niederösterreich

Preface to the Second Edition

Preparing a new edition is like a renovation. The questions are: what to keep, what to change, what is the cost, and what are the benefits?

In the present case, we kept most of the original floor plan, i.e.,
Chap. 1 (Introduction),
Chaps. 2–6 (Part I: Five Mysteries of Natural Language Communication), and
Chaps. 7–11 (Part II: The Coding of Content).
Part III, however, which consisted of a lone Chap. 12 (Conclusion), was extended into the new Chaps. 12–16. The revised conceptual structure of the overall text may be shown graphically as follows:



The new Part III (From Foundations to Applications) expands on issues prepared in Parts I and II. The three full parts each have five chapters and each chapter has six sections.

The cost is 74 additional pages. The benefits are an account of how DBS evolved (Chap. 12), a comparison of different forms of semantics (Chap. 13), examples showing similarities and differences between the natural prototype and the artificial model (Chap. 14), a discussion of corpus linguistics (Chap. 15), and an outlook on the interaction between theory and practice in DBS (Chap. 16). Also, Chap. 1 was revised using part of the original Chap. 12, the second half of Chap. 8 was replaced with new text, and left-over text from the revised Chaps. 1, 8, and 12 was absorbed into the new Part III.

Other than that, errors like faulty cross-references have been fixed, the bibliography and the indices have been adapted, some terminology has been tightened, and intensive prose pruning has been done throughout: what was almost impossible to put on the page the first time, fell often effortlessly into place with the benefit of hindsight. Questions which arose from suboptimal formulations in the 1st Ed. are hopefully answered at least in part by the 2nd Ed.

May 2015 Roland Hausser

Preface

Database Semantics (DBS) models the cycle of natural language communication as a talking robot which can switch between the hear mode, the think mode, and the speak mode. The hear mode maps language-dependent surfaces (input) into cognitive content which is stored in the agent's memory (data base). The think mode (i) selectively activates content by navigating along the semantic relations between elementary contents and (ii) derives new content from activated content by matching the antecedent (induction) or the consequent (abduction) of inferences. In the speak mode, lexicalization rules realize language-dependent surfaces from activated content (output).

A content is represented as a set of nonrecursive (flat) feature structures with ordered attributes, called proplets (because they are the elementary items of propositions). The semantic relations of structure connecting the proplets in a content are defined by address. This makes proplets order-free, which is essential for accommodating the storage and retrieval mechanism of the agent's content-addressable database.

A content proplet is turned into a pattern by replacing one or more constant values with variables. The computational matching between pattern and content proplets is efficient because proplets are nonrecursive and their attributes are ordered. Pattern matching is used in the application of operations to content, the extraction of information from memory, and the definition of reference as a relation between the language and the context level in the agent's cognition.

An important class of values are concepts which are defined as elementary recognition and action procedures of the agent (grounding). Just as the concepts of the context level are reused as elementary meanings at the language level, the compositional semantics and the word form classification of the language level are reused for composition at the context level. In this way, the context level is basically constructed from the language level by omitting the language-dependent surfaces[1] of the language proplets (4.3.3).

[1] For example, dog, chien, Hund, and cane are different surfaces for the same concept (6.6.3).

The export of language-level constructs to the context level makes the interaction between the two levels during communication simple and direct. All non-literal use, e.g. metaphor, is based on context-level inferencing (think mode) before the utterance in the speak mode and after in the hear mode.

Chapter 1 describes the level of abstraction at which a natural and an artificial cognitive agent may be viewed as functionally equivalent and lists the properties which all natural languages have in common (universals). After recapitulating the distinction between the literal meaning of a language sign and the speaker meaning of an utterance, the formal nature of a language and a nonlanguage *content* is introduced.

Chapter 2 presents a detailed discussion of *modality conversion*. One kind consists in converting an unanalyzed agent-external surface into an agent-internal modality-free content in the hear mode and the inverse mapping in the speak mode (Sect. 2.2). Another kind consists in interpreting signs in one modality and producing them in another (Sect. 2.3), as in reading aloud (vision to speech) and taking dictation (speech to vision).

Chapter 3 explains the systematic derivation of patterns from contents and their matching for functor-argument (3.2.2, 3.2.3) and coordination (3.2.5, 3.2.6). The order-free nature of proplets, based on coding the semantic relations between them as proplet-internal addresses, is illustrated in 3.2.8. Sect. 3.6 relates DBS to the dichotomy between linguistic relativism and universal grammar.

Chapter 4 relates the data structure of proplets and the database schema of a content-addressable word bank (4.1.1) to the basic mechanism of natural language communication (4.1.2). The mechanism is extended to question answering (Sect. 4.2), reference (Sect. 4.3), and coreference (Sect. 4.4). The chapter concludes with the component structure of a DBS agent, which shows the relation between the input component, the database (memory), the rule component, and the output component.

Chapter 5 describes the interaction between the cognitive agent's memory (database) and autonomous control. In contradistinction to a conventional database, which is controlled by the commands of an external user, a word bank is controlled by the agent-internal principle of balance. The technical means are inferences which take an evaluated content as input and produce a blue print for action (countermeasure) as output.

Chapter 6 explores the transition from fixed behavior agents to agents capable of learning, i.e. agents which are able to turn experiences into inferences. Agents capable of coordination only are extended to agents which are also capable of functor-argument, similar to the transition from propositional to predicate calculus in symbolic logic (Sect. 6.4). Sect. 6.5 shows how the inferences of DBS may reconstruct the subset hierarchies of the substitution approach.

Chapter 7 develops a graphical representation for the semantic relations of subject/predicate, object\predicate, modifier|modified, and conjunct−conjunct at the phrasal, elementary, and clausal levels of grammatical complexity. In the hear mode, function words are absorbed into the proplets of the resulting content and reconstructed in the speak mode by function word precipitation. The automatic generation of graphical representations from corresponding contents defined as sets of proplets is explored.

Chapter 8 investigates how nonlanguage contents may be constructed from nonlanguage recognition without the time-linear derivation order provided by the incoming surfaces of the hear mode. The results are formalized as a DBS.Content grammar (8.2.2). The computational complexity of DBS is considered from the viewpoint of recursive ambiguity and determined to be linear. The chapter concludes with integrating the RBC (recognition by components) theory of cognitive psychology into DBS.

Chapter 9 introduces the graph-theoretic concepts of a node's degree and of a graph's degree sequence, and relates them to the semantic relations graphs of DBS. Intralanguage paraphrase (Sect. 9.2) is analyzed in terms of alternative traversals of the same NAG (numbered arcs graph) and extended to extralanguage paraphrase (Sect. 9.3). Paraphrase is distinguished from other semantic relations (Sect. 9.4) and related to the dichotomy between marked and unmarked grammatical structures (Sect. 9.5).

Chapter 10 uses a fictional conversation between Jean-Paul Sartre and Simone de Beauvoir in the year 1930 for analyzing the elementary dialogs of (i) statement, (ii) WH question and (iii) Yes/No question with answer, and (iv) request with fulfillment in the speak and the hear mode. A central notion is the STAR, with the different functions of the STAR-0, STAR-1, and STAR-2 for (1) anchoring a content and for (2) the interpretation of indexicals, in the speak and the hear mode.

Chapter 11 turns from dialog to running text. The focus is on the indexical vs. coreferential interpretation of 3rd person pronouns depending on their position in clausal (extrapropositional) functor-arguments. The Langacker-Ross constraint is explained in terms of the *accessibility* of an antecedent/postcedent during the traversal (selective activation) of a semantic structure. The explanation extends to the classic examples of the Donkey sentence and the Bach-Peters sentence.

Chapter 12 provides a step by step survey of how DBS evolved from the attempt to reinterpret Montague grammar as the hear mode of a cognitive agent. Benchmarks are the move from a sign-based to an agent-based approach, the concomitant replacement of truth conditions with the agent's elementary recognition and action procedures (grounding), and the distinction between the hear, the think, and the speak mode. The most basic point is the change from a [-sense, -constructive] to a [+sense, +constructive] ontology (FoCL Sect. 20.4).

Chapter 13 describes how the move from a sign-based truth-conditional semantics to an agent-based grounded semantics resulted in the new database concept of a now front. It is illustrated with (i) a detailed hear mode derivation as it takes place at the now front, (ii) the selective activation by data at the current now front of content stored in the word bank's sediment, and the use of coactivated content for inferencing taking place at the now front.

Chapter 14 compares a Cartesian description of spatio-temporal location with an agent's on-board spatio-temporal orientation without any high-tech support. Further topics are the functional flow from recognition to action, data correction, and the avoidance of memory overflow.

Chapter 15 turns to corpus linguistics and proposes the systematic monitoring of a language by means of a Reference, Monitor, Domain corpus. An RMD corpus is based on the annual construction of monitor corpora which resemble the initial reference corpus in overall size, choice of domains, domain sizes, etc. The monitor corpora are built systematically from the same basket of renewable language sources such as long-running newspapers for everyday language and established journals from different domains. The formal analysis of corpus data in a DBS corpus word bank is illustrated in detail.

Chapter 16 concludes with the practical application of machine translation and a typology of DBS operations.

Remark on Footnotes and References

The text contains many bibliographical references and footnotes. These are intended for readers who want to explore a topic in greater depth. The casual reader may just ignore the references and footnotes, and stick to the text alone.

As an eBook, the text provides automatic cross-referencing. A mouse click on a reference number opens the page in question. A click on the go and then the back button provided by the top menue bar in Adobe pdf brings the reader back to the original page.

Website for vita, list of publications, and online papers: `lagrammar.net`

The titel of this book is abreviated with the acronym CLaTR.

Abbreviations Referring to Preceding and Subsequent Work

Preceding Work

SCG Hausser, R. (1984) *Surface Compositional Grammar*, pp. 274, München: Wilhelm Fink Verlag

NEWCAT Hausser, R. (1986) *NEWCAT: Natural Language Parsing Using Left-Associative Grammar*, (Lecture Notes in Computer Science 231), pp. 540, Springer

CoL Hausser, R. (1989) *Computation of Language: An Essay on Syntax, Semantics and Pragmatics in Natural Man-Machine Communication*, Symbolic Computation: Artificial Intelligence, pp. 425, Springer

TCS Hausser, R. (1992) "Complexity in Left-Associative Grammar," *Theoretical Computer Science*, Vol. 106.2:283–308, Amsterdam: Elsevier

FoCL Hausser, R. (1999) *Foundations of Computational Linguistics, Human-Computer Communication in Natural Language, 3nd ed. 2014*, pp. 518, Springer

AIJ Hausser, R. (2001) "Database Semantics for Natural Language." *Artificial Intelligence*, Vol. 130.1:27–74, Amsterdam: Elsevier

L&I'05 Hausser, R. (2005) "Memory-Based Pattern Completion in Database Semantics," *Language and Information*, Vol. 9.1:69–92, Seoul: Korean Society for Language and Information

NLC Hausser, R. (2006) *A Computational Model of Natural Language Communication – Interpretation, Inferencing, and Production in Database Semantics*, Springer, pp. 360;
2nd Ed. 2017, pp. 363, preprint at `lagrammar.net`

L&I'10 Hausser, R. (2010) "Language Production Based on Autonomous Control - A Content-Addressable Memory for a Model of Cognition," in *Language and Information*, Vol. 11:5-31, Seoul: Korean Society for Language and Information

Subsequent Work

L&I'15	Hausser, R. (2015) "From Montague grammar to Database Semantics," in *Language and Information*, Vol 19(2):1-18, Seoul: Korean Society for Language and Information
HBTR	Hausser, R. (2017) *How to build a Talking Robot – Linguistics, Philosophy, and Artificial Intelligence*, pp. 120, Springer Verlag
TExer	Hausser, R. (2017) *Twentyfour Exercises in Linguistic Analysis. DBS software design for the hear and the speak mode of a talking robot*, pp. 316, preprint of Chapter 1 at `lagrammar.net`

References are to (preprints of) the latest editions.

Contents

1. Introduction: How to Build a Talking Robot

Can computational linguistics steer clear of artificial cognitive agents with language, i.e. talking robots? The answer is *no* if the research goal of our interdisciplinary field is a functional reconstruction of free natural language communication, verified by a computational model. The practical outcome of this approach is unrestricted human-machine communication in natural language.

A talking robot requires language cognition as well as nonlanguage cognition. For example, a human telling the artificial agent what to do requires the machine to understand natural language and to perform nonlanguage action. Similarly, for the artificial agent to tell a human what is going on requires the machine to have nonlanguage recognition and natural language production.[1]

The task of building a talking robot is not just a metaphor, but contributes the necessity (i) to distinguish between the robot-external environment and the robot-internal cognition, (ii) to provide interfaces as a link between the environment and the agent's cognition, and (iii) to build an autonomous control for connecting the robot's recognition and action in a meaningful way.[2]

We honor these plain facts by making them founding assumptions of Database Semantics (DBS). They provide valuable requirements and restrictions which differ from the founding assumptions of symbolic logic in analytic philosophy and of nativism in theoretical linguistics, yet provide excellent heuristics for designing the functional flow of the DBS software.

1.1 Level of Abstraction

The construction of an artificial agent may be compared to the construction of an airplane. For each, there is a natural prototype which they share certain crucial properties with and from which they differ in others.

An airplane (artificial model) and a bird (natural prototype) have in common that both stay airborne according to the same principles of the theory of aero-

[1] The constellations of recognition and action interacting at the language and nonlanguage levels of an agent are shown in FoCL Sect. 23.5 as the ten SLIM states of cognition.

[2] There are additional requirements, such as an agent-internal database, but these are not robot-specific.

dynamics. They differ, however, in that lift and propulsion are combined in the flapping wings of birds, but separated in planes into the fixed wing for lift and the propeller or jet engine for propulsion (FoCL, Intro. VII).

Correspondingly, a talking robot and its human prototype have in common that both connect external interfaces to a memory. They differ, however, in that the robot's cognition uses electronic hardware and the methods of computer science, while the cognition of the human prototype is based on natural wetware and eons of evolution (Chap. 14).

Orthogonal to the analogy between artificial flying and artificial cognition is a crucial difference regarding their interaction with the human user. In an airplane, the method of being airborne is completely separate from its contact areas with humans. The latter concerns the size of the doors and the seats, the cabin pressure, the service, etc., while the former concerns the shape of the wings, the manner of propulsion, the technique of takeoff and landing, etc.

In a talking robot, in contrast, the functioning of communication and the contact areas with the human user are closely related. For example, if the human uses speech, understanding by the robot requires the ability to hear, just as the interpretation of written language requires the robot's ability to see. Also the robot's response requires the production of sounds for speech and of signs for writing. In addition, the cognition of the robot must provide a memory with similar storage and retrieval capabilities, similar reasoning, and a similar interpretation of the world as the human prototype. Thus it must recognize a color, for example, and classify it like the members of its language community.

This correlation between the talking robot and its natural prototype is formulated as the *Principle of Functional Equivalence* in DBS:[3]

1.1.1 Principle of Functional Equivalence (FE)

1. The more realistic the reconstruction of natural cognition, the better the functioning of the artificial model.
2. The better the functioning of the artificial model, the more realistic the reconstruction of natural cognition.

FE is intended to ensure successful long-term upscaling. Functional equivalence[4] is defined at levels of abstraction at which it does not matter for completeness of function and of data coverage whether cognition is based on natural wetware or electronic hardware. FE applies to cognitive aspects of a talking

[3] The Principle of Functional Equivalence has been preformulated as the Equation Principle in NLC 1.3.1; it combines i. the Verification Principle (NLC Sect. 1.2), ii. the Equation Principle (NLC Sect. 1.3), iii. the Objectivation Principle (NLC Sect. 1.4), iv. the Interface Equivalence Principle (NLC 1.5.1), and v. the Input/Output Equivalence Principle (NLC 1.5.2).

agent which are (i) concretely observable and (ii) impact directly the quality of free human-machine communication in natural language.

1.2 Universals

DBS defines successful language communication succinctly as a transfer of content from the cognition of the speaker to the cognition of the hearer, solely by means of a time-linear sequence of modality-dependent unanalyzed external language surfaces (Chap. 2). The transfer is successful if the content coded by the speaker is reconstructed and stored equivalently by the hearer.

The theoretical and computational modeling of this transfer mechanism is based on the following properties, which are universal[5] among the natural languages of the world:

1.2.1 Universals of natural language communication

1. The cycle of natural language communication is based on the *hear*, the *think*, and the *speak* modes of cognitive agents.
2. In communication, expressions of natural language are interpreted relative to an agent-internal *context*.
3. All natural languages have a *time-linear* structure, i.e. linear like time and in the direction of time.
4. All natural languages use the three kinds of sign *symbol, index,* and *name*, each with its own mechanism of reference.

---

[4] There exist many examples of functional equivalence. Take for example the basic operations of arithmetic, i.e. addition, subtraction, multiplication, and division. They are defined at a level of abstraction which may be realized equivalently as the (i) natural, (ii) mechanical, and (iii) electronic operations of a human, a mechanical calculating machine, and an electronic calculator, respectively.

The functional equivalence is apparent from the fact that any operation of arithmetic, for example 5–3, will have the same result regardless of whether it is performed by a human, a mechanical calculator, or a computer. Differences regard only freedom from error, speed of operation, and maximal size of the numbers, i.e. properties which are orthogonal to the abstract mathematics.

The functional equivalence of different implementations of arithmetic is simpler than that of DBS because the operations of arithmetic may be defined independently of any agent. This is assumed for mathematical truths in general, at least in the school of *mathematical realism*, according to which the laws of mathematics hold independently of whether or not they have been discovered by humans.

[5] The differences between natural languages are comparatively minor (NLC 4.6.1). In this respect, we agree with Chomsky. However, while nativism formulates the similarity between languages as an innate Universal Grammar, DBS derives it from a common mechanism of transferring content in communication. See MacWhinney (2004) for an evaluation of the nativist universals from the viewpoint of cognitive psychology.

5. All natural languages use classic *coordination* and *functor-argument*[6] to compose content at the *elementary*, the *phrasal*, and the *clausal* level.
6. All natural languages distinguish between object (noun, argument), relation (verb, functor), and property (adj, modifier).[7]
7. All natural languages have the sentential moods *declarative, interrogative,* and *imperative*.

These universals characterize an agent-based approach,[8] and provide the building blocks for and the constraints on the design of a talking robot. Once the natural language communication mechanism has been implemented as a basic, high-level software machine based on selected examples from a natural language of convenience, here mostly English, the sample set may be extended to additional constructions and to other languages and contents (Sect. 16.2).

At the most general level, the DBS robot requires the following constructs:

1.2.2 Requirements of a grounded artificial agent

In order to be grounded, a cognitive agent requires a body with

1. *interfaces*
   for recognition and action, based on
2. a *data structure*
   for representing content,
3. a *database schema*
   specifying how content is stored and retrieved,
4. an *algorithm*
   for reading content into and out of the database as well as for processing content, such that (1–4) combine into
5. a *software machine*
   which models the cycle of natural language communication as well as language and nonlanguage inferencing.

---

[6] The semantic relations of functor-argument and coordination go back to Aristotle and are the basic relations in the predicate calculus of symbolic logic as conceived by Frege (1891). We use functor-argument instead of functor-argument *structure* because it is shorter and more proportionate to its sister, i.e. coordination.

[7] Whether all natural languages have the same basic parts of speech is discussed controversially in language typology (Sect. 3.6).

[8] An agent-based approach is in contradistinction to the sign-based approaches of today's linguistics and philosophy of language.

The software for the input and output of language as well as nonlanguage data must be suitable for connecting to the recognition and action hardware of autonomous robots when they become available in computational linguistics.

Until then, we have to make do with today's general-purpose computers. Despite their limitation to the keyboard for recognition and the screen for action as their main interfaces, they allow us to run the language interpretation, the language production, and the reasoning software. The computational implementation allows the researchers to observe the cognitive operations of the artificial agent directly (tracing), which is of great heuristic value.

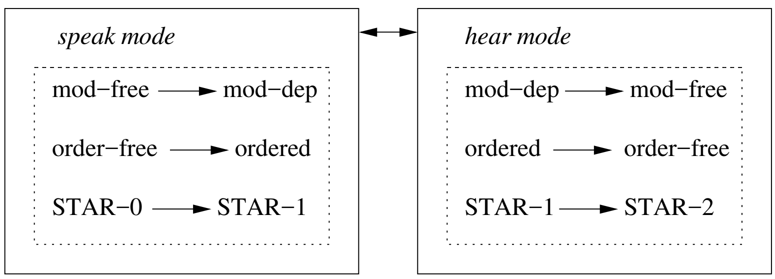
Natural language communication requires the following *conversions* into which all upscaling has to be embedded:

1.2.3 Conversion Universals of DBS

1. From the agent's *speak* mode to its *hear* mode and back (Chap. 3),
2. from a *modality-free* to a *modality-dependent* representation of the surface in the speak mode and back in the hear mode, in word form production (synthesis) and (automatic word form) recognition (Chap. 2),
3. from *order-free content* to *ordered surfaces* in the speak mode and back in the hear mode (Chap. 3), and
4. from the STAR-0 to the STAR-1 perspective in the speak mode and from the STAR-1 to the STAR-2 perspective in the hear mode (Chaps. 10, 11).[9]

Given that each kind of conversion is realized in two directions, there are altogether eight. Thereby, the conversions 2–4 are embedded into conversion 1 between the speak mode and the hear mode, as shown graphically below:

1.2.4 Internal structure of the DBS Conversion Universals



The Conversion Universals combine into a fundamental framework of structure and function, which forms the backbone of the DBS software machine.

[9] A STAR represents the parameters of Space, Time, Agent, and Recipient (Chap. 10).

1.3 Computational Verification

The long history of the natural sciences provides many examples of theories which sooner or later revealed some deep-seated flaw. In such a case, the theory cannot be fixed without completely uprooting its conceptual and, if present, mathematical framework, and in consequence much of its empirical work, which is unfortunate. When such a flaw has been found, it may take decades or even centuries until it is widely recognized and repaired.

In modeling the communication cycle, there is a similar process of revealing deep-seated flaws, though much faster. It is based on the method of *upscaling* the cognition software *incrementally*. At each stage, automatic testing on large amounts of relevant data reveals remaining deficiencies of function and of data coverage. These are so painfully evident to the research group that questions of correctness can be decided without delay. Remaining controversies may be left to the next stages of upscaling until a clearer picture appears.

With completeness of function and of data coverage as its long-term goal, DBS uses success and failure of incremental upscaling as its method of computational verification (NLC Sect. 1.2):

1.3.1 PRINCIPLE OF COMPUTATIONAL VERIFICATION (CV)

> If the long-term upscaling of the communication cycle runs into problems, then (some part of) the theory must be corrected; conversely, if upscaling succeeds, the theory is on the right track (for now).

The principle of CV differs from the verification principles of the natural sciences (repeatability of experiments) and mathematics (proof of consistency), though it is compatible with them. That computational verification is a highly effective method is shown by the development of DBS, which has seen its full share of major revision and correction (Chap. 12).

For long-term upscaling, a well-designed implementation is a methodological necessity because without it the relative correctness of the theory cannot be tested automatically (and thus objectively, reliably, and efficiently) on massive amounts of data. After implementations in Lisp[10] and C, DBS is currently implemented as the JSLIM software written in Java.

Of course, we can only hope that the design of DBS will sustain the upscaling effort long-term. We are optimistic, however, because the cognitive

[10] Thanks to Stanley Peters and the CSLI for the 1984–1986 stay at Stanford University. By generously providing the then newest technology, CSLI made programming the first left-associative parser possible, initially in Maclisp on a DEC Tops-20 mainframe, then in Interlisp-D on Xerox Star 8010 Dandelion and Dandetiger workstations, and finally in Common Lisp on an HP 9000 Bobcat.

distinctions and building blocks listed in 1.2.1 – 1.2.4 are so general, obvious, and simple that an alternative model of natural language communication doing without them is likely to have serious problems with incremental upscaling.

1.4 Literal Meaning$_1$ and Utterance Meaning$_2$

Beyond the universals listed in 1.2.1 and the CV principle 1.3.1, the foremost foundation of the DBS approach to natural language communication is the distinction between the literal meaning$_1$ of a language sign and the speaker meaning$_2$ of an utterance. The functional interaction between these two notions of meaning is formulated as the First Principle of Pragmatics (FoCL 4.3.3.):

1.4.1 FIRST PRINCIPLE OF PRAGMATICS (POP-1)

> The speaker's utterance meaning$_2$ is the use of the sign's literal meaning$_1$ relative to an agent-internal context of interpretation.

The use of a sign's literal meaning$_1$ in an utterance requires a cognitive agent. Therefore PoP-1 implies an agent- rather than a sign-based approach from the outset. An agent-based approach in turn requires a distinction between the (i) agent-external environment and the (ii) agent-internal cognition (NLC 1.4.3). The connection between the two is provided by the external interfaces for recognition and action, an agent-internal memory, and an algorithm mapping content between memory and the external interfaces.

PoP-1 relates to a controversy (Pelletier 2001) as to how Frege's use of the term *meaning* should be interpreted. Consider the use of the term in the Fregean principle (FoCL 4.4.1):

1.4.2 THE FREGEAN PRINCIPLE

> The meaning of a complex expression is a function of the meaning of its parts and their mode of composition.

Assuming PoP-1, the controversy boils down to the question of whether this seminal principle of natural language semantics should be applied to the sign's literal meaning$_1$ or to the speaker meaning$_2$ of an utterance. The second option is faced with arbitrarily many counter-examples involving non-literal uses, while the first allows the Fregean Principle to hold without exception.

By restricting the Fregean Principle to the sign's literal meaning$_1$ it may be turned into the methodological principle of surface compositionality:[11]

[11] SCG; FoCL; NLC.

1.4.3 Surface Compositionality

> An analysis of language signs is surface compositional if it uses only concrete word forms as building blocks, such that all syntactic and semantic properties of a complex expression derive systematically from (i) the lexical properties of the isolated word forms and (ii) their syntactic-semantic composition as functor-argument and coordination.

This moves all aspects of natural language use (including metaphor, collocation, idiom, and the like) into the pragmatics, where they belong, and excludes from the syntactic-semantic analysis any operations known to increase complexity to exponential or undecidable, e.g. transformations.[12]

1.5 Declarative Specification

In any running computer program, two basic aspects must be clearly distinguished: (i) the necessary (or essential) properties of the system, to be described by a *declarative specification*,[13] and (ii) the accidental[14] properties of the *procedural implementation*.

Accidental properties of an implementation (program) are the choice of the hardware and the programming language, as well as idiosyncrasies of the programmer. Necessary properties, in contrast, are the interfaces, the data structure, the algorithm, the component structure, the functional flow from input to output, and the database schema of the overall system. It follows that one declarative specification may have an open number of different implementations which are all equivalent with respect to the necessary properties (NLC 1.2.1).[15]

---

[12] A functional counterpart to the methodological principle of surface compositionality is the First Mechanism of Communication (MoC-1, 2.2.2).

[13] It is possible for a computer program to run successfully without an explicit declarative specification. In such a case, it is a matter of additional work to abstractly state which properties are intended to be necessary.

[14] The term accidental is used here in the philosophical tradition of Aristotle (384–322 B.C.), who distinguishes in his Metaphysics, Books ζ and η, between the necessary and the accidental (incidental or coincidental – kata sumbebêkos) properties of an object in nature.

[15] The dichotomy between a declarative specification and its possible procedural implementations differs from the nativist dichotomy between competence (language knowledge) and performance (processing knowledge). The latter distinction is used for "competence grammars" intended to model an innate human language ability, reified as a language acquisition device (LAD).

Chomsky, Sag, and others have emphasized repeatedly that competence grammars are *not intended* to model the language processing by the speaker-hearer. Therefore it is not surprising that the functional flow of competence grammars (NLC 3.4.3) is incompatible with that of a talking robot. Also, competence grammars are cumbersome to program because they fail to be *type-transparent* (Berwick

In DBS, an essential part of the declarative specification is the choice of the basic data structure – analogous to the basic units of other sciences such as the cell in biology or the atom in chemistry and physics. The basic unit of DBS is the *proplet*, defined as a flat feature structure[16] with ordered attributes taking lists of n (n≥0) atomic items as their values.

Proplets turn out to be versatile in that they maintain their format and their formal properties in a multitude of different functions. Consider the following examples of a language proplet for German, a language proplet for French, a context proplet, and three pattern proplets (for more examples see Sect. 6.6):

1.5.1 Comparison of language, context, and pattern proplets

German proplet	*French proplet*	*context proplet*	*pattern proplet1*	*pattern proplet2*	*pattern proplet3*
sur: Hund	sur: chien	sur:	sur: Hund	sur: α	sur:
noun: dog	noun: dog	noun: dog	noun: dog	noun: dog	noun: β
cat: m-g	cat: sn	cat: sn	cat: m-g	cat: m-g	cat: sn
sem: sg	sem: m sg	sem: sg	sem: sg	sem: sg	sem: sg
fnc: bark	fnc: bark	fnc: bark	fnc: bark	fnc: bark	fnc:
mdr: old	mdr: old	mdr: old	mdr: old	mdr: old	mdr:
prn: 23	prn: 24	prn: 25	prn: K	prn: K	prn: K
1	2	3	4	5	6

The above proplets share the core attribute, noun. Proplets 1–5 also share the core value dog. Proplets 1, 2, and 3 are content proplets because they do not contain any variable, in contradistinction to the pattern proplets 4, 5, and 6. Proplets for a word in different languages differ mainly in their sur(face) value, here 1 Hund vs. 2 chien. Language and context proplets differ in the presence or absence of a sur value (1, 2 vs.3). Pattern proplet 4 with the prn variable K matches all language proplets equivalent to 1, pattern proplet 5

and Weinberg 1984; FoCL Sect. 9.3). declarative specification of DBS, in contrast, is an entirely computational theory satisfying absolute type transparency (FoCL Sect. 10.4).

A declarative specification differs also from an *algebraic definition* in logic. Based solely on set theory, an algebraic definition is neither equipped nor intended to represent the higher level notions of a computer program. What is missing is the specification of a functional flow from input to output.

As a means to provide complex software systems with declarative specifications, high level modeling languages like UML (Unified Modeling Language, e.g. Fowler 2004) and ER (Entity Relationship, e.g. Chen 1976, Thalheim 2000, Beynon-Davies 2003) have been designed in computer science. These formal languages are, however, not flexible and detailed enough to define all the software constructs of DBS – though analyzing certain aspects of DBS in UML, ER, or other projects of similar intent could well be worthwhile.

[16] A precursor of feature structures in linguistics is a list of binary values without attributes, called a "feature bundle", e.g. $\begin{bmatrix}\text{+vocalic}\\ \text{+high}\end{bmatrix}$, used by Chomsky and Halle (1968) for purposes of morphophonology. The introduction of feature structures with attribute-value pairs is credited to Minsky (1975). Called "frames" by Minsky, their origin may be traced back to the legendary Dartmouth workshop of 1956, attended also by Newell, McCarthy, Rochester, Shannon, and Simon.

with the prn variable K and the sur variable α matches all language proplets equivalent to 1 and 2, and pattern proplet 6 matches all singular noun proplets.

In addition, there is the distinction between lexical proplets and connected proplets. The latter represent content resulting from what is called *encoding* in database applications of computer science, here by means of hear mode derivations (3.3.1), nonlanguage recognition (Chap. 8), and inferencing (Chaps. 5, 6, 10, 11).

A crucial operation supported by the data structure of proplets is pattern matching between pattern proplets and content proplets. A pattern proplet may have several variables and don't care values represented by empty space, e.g. [mdr:]. The matching between pattern and content proplets is successful, if they satisfy (i) the attribute condition and (ii) the value condition (NLC 3.2.3).

The attribute condition requires that the attributes in the pattern proplet must be equal to, or a sublist of, the attributes in the content proplet. The value condition requires that corresponding values must be compatible. Consider the following examples of pattern matching in different kinds of operations:[17]

1.5.2 Operations based on pattern matching

$$
\begin{array}{llll}
 & \textit{rule pattern} & \textit{language proplet} & \textit{retrieval pattern} \\
 & \textit{matching content} & \textit{matching context} & \textit{matching data} \\[6pt]
\textit{pattern} &
\begin{bmatrix} \text{sur:} \\ \text{noun: } \alpha \\ \text{cat:} \\ \text{sem: sg} \\ \text{fnc: } \beta \\ \text{mdr:} \\ \text{prn: K} \end{bmatrix} &
\begin{bmatrix} \text{sur: Hund} \\ \text{noun: dog} \\ \text{cat: m-g} \\ \text{sem: sg} \\ \text{fnc: bark} \\ \text{mdr: old} \\ \text{prn: 23} \end{bmatrix} &
\begin{bmatrix} \text{sur:} \\ \text{noun: } \beta \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{prn: K} \end{bmatrix} \\[6pt]
\textit{content} &
\begin{bmatrix} \text{sur: Hund} \\ \text{noun: dog} \\ \text{cat: m-g} \\ \text{sem: sg} \\ \text{fnc: bark} \\ \text{mdr: old} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 23} \end{bmatrix} &
\begin{bmatrix} \text{sur:} \\ \text{noun: dog} \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc: bark} \\ \text{mdr: old} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: c23} \end{bmatrix} &
\begin{bmatrix} \text{sur: Hund} \\ \text{noun: dog} \\ \text{cat: m-g} \\ \text{sem: sg} \\ \text{fnc: bark} \\ \text{mdr: old} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 23} \end{bmatrix} \\
 & \quad 1 & \quad 2 & \quad 3
\end{array}
$$

[17] In the JSLIM software (Weber and Handl 2010), proplets always have all their attributes in the appropriate order (NLC Sect. A.2). Here, however, proplets are handled in a more liberal fashion: they are often shown with only a sublist of their attribute-value pairs. The purpose is a more succinct presentation and drawing attention to properties relevant for the issue at hand.

The pattern matching in these examples is successful. The attribute condition is fulfilled because the attributes in the patterns equal the attributes in the contents. Compatibility between values is fulfilled as follows.

In 1, the values relevant for matching, i.e. the variables α, β and K in the pattern proplet, are compatible with the corresponding constants dog, bark, and 23 in the content proplet (no variable restrictions). In 2, the value relevant for matching, i.e. dog as a concept *type* in the language proplet, is compatible with the corresponding *token* in the context proplet (4.3.3). In 3, the pattern proplet is underspecified[18] (generalized) to a degree that it matches all singular common nouns in a suitable database (word bank, 4.1.1).

Compared to the recursive feature structures of nativism (with unordered attributes, but an order of embedding), proplets have the following advantages:

1.5.3 Advantages of proplets

1. Flat ordered feature structures are easier to read and computationally more efficient than recursive feature structures with unordered attributes.
2. Flat ordered feature structures provide for easy pattern derivation and for easy pattern matching.
3. The combination of a proplet's core and prn value provides a natural primary key for storage in and retrieval from memory.
4. Coding interproplet relations as addresses (Sect. 4.4) makes proplets order-free and therefore amenable to the needs of one's database.
5. A time-linear navigation along interproplet relations reintroduces order and is used for selective activation, inferencing, and language production.

Internally a proplet consists of features which are *ordered*, as in a list. Each feature is defined as a flat (non-recursive) attribute-value pair (avp). Proplets as non-recursive feature structures with ordered attributes are the simplest and most efficient data structure for coding the properties of elementary contents, e.g. word forms.

Externally, in contrast, *unordered* proplets (3.2.7) are connected into contents of arbitrary size. This is based on defining interproplet relations by means

[18] The underspecification of content provided by pattern proplets is distinct from the "semantic underspecification" of Minimal Recursion Semantics (MRS) proposed by Copestake, Flickinger, Pollard, and Sag (2006). MRS is intended to reduce the proliferation of artificial ambiguities caused by alternative quantifier scopes in predicate calculus (13.2.1), though without providing a cure for failing quantifier scope, as in the donkey problem (11.4.5).

Database Semantics, in contrast, has neither quantifiers nor quantifier scope (13.2.2). Instead, under-specification is used for pattern matching. Patterns are automatically derived from content (Sects. 3.2, 6.5). Pattern matching is essential for the application of operations, retrieval, query answering, and so on.

of addresses stored inside the proplets. Coding a complex content as a set of order-free proplets connected by address allows to accommodate its storage and retrieval in any database schema.

1.6 Language, Content, and Thought

The component structure and functional flow of a cognitive agent with language has been described as the SLIM[19] theory of language (FoCL Intro. XI, NLC Sect. 2.6). SLIM is an agent-based approach which provides the theoretical foundation of the DBS software machine for a talking robot.

SLIM and DBS share the fundamental assumption that language content and context (nonlanguage) content should be coded and processed essentially the same (6.4.4).[20] This applies to the data structure, the algorithm, the storage and activation (retrieval) in a content-addressable database, the functor-argument and coordination relations, the use of the type-token distinction for pattern matching, and the reference mechanisms of concepts, pointers, and baptism.

The incremental growth of a content in the hear mode may be shown as follows (FoCL 5.4.1):

1.6.1 SIMPLIFIED HEAR MODE DERIVATION

surface	⇒ The_dog	⇒ found	⇒ a_bone
	⇓	⇓	⇓
language level	[sur: The_dog noun: dog cat: snp sem: def sg fnc: find mdr: nc: pc: prn: 23]	[sur: found verb: find cat: #n′ #a′ decl sem: past arg: dog bone mdr: nc: pc: prn: 23]	[sur: a_bone noun: bone cat: snp sem: indef sg fnc: find mdr: nc: pc: prn: 23]
	⇓	⇓	⇓
context level	[sur: noun: dog cat: snp sem: def sg fnc: find mdr: nc: pc: prn: 23]	[sur: verb: find cat: #n′ #a′ decl sem: past arg: dog bone mdr: nc: pc: prn: 23]	[sur: noun: bone cat: snp sem: indef sg fnc: find mdr: nc: pc: prn: 23]

[19] The acronym SLIM abbreviates **S**urface compositional, time-**L**inear, **I**nternal **M**atching, which are the methodological, empirical, ontological, and functional principles, respectively, of the approach. When the acronym is viewed as a word, SLIM indicates low mathematical complexity, resulting in the computational efficiency required for real-time processing.

[20] The guiding principle is *form follows function* (Cuvier 1817/2009).

The hear mode interpretation takes a time-linear sequence of modality-dependent, unanalyzed, external surfaces as input (⇒). The arrival order of the surfaces is used to derive (⇓) an order-free content, by automatic word form recognition and syntactic-semantic concatenation at the language level, and by deleting the sur values, resulting in context proplets.

The corresponding speak mode derivation uses the proplets derived in the hear mode 1.6.1, starting at the level of context (FoCL 5.4.2):

1.6.2 Simplified Speak Mode Derivation

surface		The_dog		found		a_bone
		⇑		⇑		⇑
language level		sur: The_dog noun: dog cat: snp sem: def sg fnc: find mdr: nc: pc: prn: 23		sur: found verb: find cat: #n′ #a′ decl sem: past arg: dog bone mdr: nc: pc: prn: 23		sur: a_bone noun: bone cat: snp sem: indef sg fnc: find mdr: nc: pc: prn: 23
		⇑		⇑		⇑
context level	⇒	sur: lexnoun(dog) noun: dog cat: snp sem: def sg fnc: find mdr: nc: pc: prn: 23	⇒	sur: lexverb(find) verb: find cat: #n′ #a′ decl sem: past arg: dog bone mdr: nc: pc: prn: 23	⇒	sur: lexnoun(bone) noun: bone cat: snp sem: indef sg fnc: find mdr: nc: pc: prn: 23

The production is driven by a navigation along the semantic relations between proplets at the context level (⇒). This is one form of thought, called selective activation of content in the agent's memory (database). The other is inferencing (Sect. 5.2), which derives new content from activated content.

The navigation at the context level reintroduces a time-linear order, lost during the hear mode derivation 1.6.1. Simultaneously, the context proplets are mapped vertically (⇑) into language proplets by constructing language-dependent sur values (3.4.3). These are realized agent-externally as modality-dependent, unanalyzed, external surfaces (automatic word form production).

The proplets at the language and the context level differ solely in the presence vs. absence of sur values. Otherwise, they use (i) the same list of attributes, (ii) the same core values, namely dog, find, and bone, (iii) the same cat and sem values, and (iv) the same continuation values, resulting in the same semantic relations of structure.

Transferring 1.6.1 and 1.6.2 to the German language would require no more than using other language-dependent sur values and a slight adjustment of the cat and sem values. If applied to the Korean language, the speak mode would also have to introduce another time-linear order, namely $noun_{NOM}$ $noun_{ACC}$ *verb*, in addition to using different sur and suitable cat and sem values. Thus, as long as the lexicalization, especially the valency structure, of two languages is similar, the language-dependent adjustments affect the representation of content only minimally.

The computational mapping from surfaces to content (hear mode), the processing of content (think mode), and the mapping from content to surfaces (speak mode), intertwines language and nonlanguage cognition as follows:

1.6.3 Interrelation of language and nonlanguage cognition

> On the one hand, language cognition uses the same external interfaces for recognition and action as nonlanguage cognition. On the other hand, nonlanguage cognition uses the same component structure and functional flow as the cycle of natural language communication.

While the hear mode takes concrete external surfaces as input, the speak mode raises the question of where its input should come from. As a first step, language production has been divided into choosing (i) *what to say* and (ii) *how to say it* (McKeown 1985; Kass and Finin 1988). In 1.6.2, the ⇒ direction at the context level constitutes the *what to say*, while the ⇑ direction constitutes the *how to say it*.

Just as the *what to say* in language action corresponds to the *what to do* in nonlanguage action, the *how to say it* corresponds to the *how to do it*. These require an autonomous control. To be effective, it should be adaptable to unexpected changes in the agent's ecological niche (and not a rigid decision mechanism predefined for some application of a repetitive nature). In DBS, autonomous control is based on the dynamic principle of balance (NLC 3.5.6), and not on the static principle of truth. Maintaining a state of balance requires executing a multitude of smaller tasks in harmony – with inaction as the limiting case.

To arrive at an idea of the functionalities demanded by the computational model of a cognitive agent with language, the following Part I is organized around five *Mysteries of Natural Language Communication*. Each mystery is answered with a Mechanism of Communication, called MoC-1–MoC-5.

Part I

Five Mysteries of Natural Language Communication

2. Mystery Number One: **Using Unanalyzed External Surfaces**

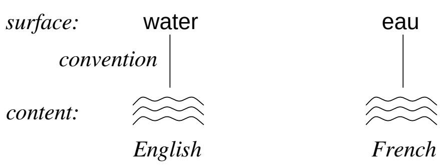
The first mystery of natural language communication may best be illustrated with a foreign language situation. For example, if our hometown is in an English-speaking country we can go to a restaurant there and successfully order a glass of water by saying to the waiter Please bring me a glass of water. If we travel to France, however, we will not be understood unless we use French or the waiter speaks English.[1]

2.1 Structure of Words

The difference between speaking in our language at home and abroad is caused by the composite structure of words. Essential components are (i) the *surface*, (ii) the *content*, and (iii) a *convention* connecting the surface and the content. In DBS, a content attached to a language-dependent surface is called a meaning$_1$.

As informal examples, consider the following analyses of the English word water and its French counterpart eau:

2.1.1 INFORMAL EXAMPLES SHOWING BASIC WORD STRUCTURE



The two words have the same content (literal meaning$_1$), represented by three wavy lines suggesting water. Their surfaces, however, are the different letter sequences water and eau. Each surface is connected to its meaning by a convention which every speaker of English or French has to learn.[2]

[1] "Whereas the individuals of all nonhuman species can communicate effectively with all their conspecifics, human beings can communicate effectively only with other persons who have grown up in their same linguistic community – typically, in the same geographical region." Tomasello (2003), p. 1.

[2] The Swiss linguist Ferdinand de Saussure (1857–1913) describes the convention-based and therefore "unmotivated" relation between language-dependent surfaces like water or eau and their content in his *Premier Principe: l'arbitraire du signe* (Saussure 1916/1972; FoCL Sect. 6.2).

The abstract type[3] of a word form surface may be realized as tokens in different modalities. In spoken language, the surface tokens are sounds which are produced by the speaker's mouth and recognized by the hearer's ears. In written language, the surface tokens are letter sequences which are produced by the speaker's (writer's) hands and recognized by the hearer's (reader's) eyes.[4]

It follows from the basic structure of words that first language acquisition requires the child to learn (i) the contents, (ii) the (acoustic) surfaces, and (iii) the conventions which connect the contents to the correct surfaces.[5] This process is embedded in child development, takes several years, and normally does not cause any special difficulties. However, when learning the words of a foreign language as an adult, the following difficulties stand out:

2.1.2 Tasks of learning the words of a foreign language

- learning to recognize and produce the foreign surfaces in the modalities of spoken and written language, and
- learning the conventional connections between the foreign surfaces and contents familiar from one's first language.

Learning to recognize the acoustic surfaces of a foreign language is difficult and to pronounce them without accent is often nearly impossible, while learning to read and to write may come easier. There are languages like Japanese, however, for which a Westerner is considered more likely to learn to speak fairly fluently than to acquire a near-native ability to read and write.

Connecting the foreign surfaces to familiar contents presupposes that the notions of the foreign language are identical or at least similar to those of one's own. This holds easily for basic notions such as *father, mother,*[6] *child, son,* and *daughter*, as well as *sun, moon, water, fire, stone, meat, fish, bird,* and *tree*. When it comes to more culturally dependent notions, however, what is represented by a single word (root) in one language may have to be paraphrased by complex constructions in the other.

---

[3] The type-token distinction was introduced by C. S. Peirce (CP 4:537). See Sect. 8.5; NLC Sect. 4.1; FoCL Sect. 3.3.

[4] A third modality is signed language for the hearing impaired. For the blind there is Braille as a form of written language.

[5] In addition, the child has to learn the syntactic-semantic composition of words into complex expressions (sentences) with complex meanings and their use conditions.

[6] It has been claimed that "in some languages" the word meaning of mother would include maternal aunts. However, the distinction between a mother's own children and those of her sisters is unlikely to be lost. A more plausible explanation is a nonliteral use. In German, for example, a child may call any female friend of the parents Tante (aunt) with familiar du. If needed, this use may be specified more precisely as Nenn-Tante (aunt by name or by courtesy). Similarly in Korean, where the term "older brother" may be bestowed on any respected older male (Kiyong Lee, personal communication).

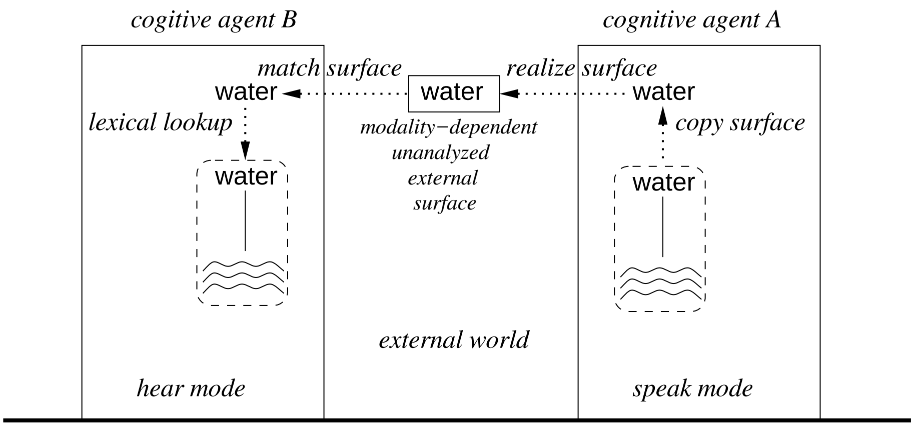
A popular example is the Inuit language, said to have something like fifty different words for snow (Boas 1911). In English translation, these would have to be paraphrased, for example, as soft snow, hard snow, fresh snow, old snow, white snow, grey snow, and snow pissed on by a baby seal.[7] Preconditions for adequate paraphrasing are a proper knowledge of the foreign notion to be described and of the notions in the native target language suitable for the task. This holds, for example, for an Alaska native faced with the task of introducing the novel notion of an electric coffee grinder into Inuit.

2.2 Modality-Dependent Unanalyzed External Surfaces

As an agent-based approach, DBS provides two basic perspectives from which communicating with natural language may be analyzed, namely *internal* and *external*. The internal (or endosystem) perspective is based on a native speaker's introspection regarding, for example, successful and unsuccessful experiences of communicating. The external (or exosystem) perspective is that of a scientist (NLC 1.4.3) working to reconstruct the language communication mechanism as an abstract theory, verified by an implementation as an artificial agent with language (talking robot).

The following example illustrates the external perspective with two agents, A and B. A is in the speak mode and produces the modality-dependent unanalyzed external word form surface water. B is in the hear mode and recognizes this surface.[8]

2.2.1 PRODUCTION AND RECOGNITION OF A WORD



[7] See Pullum (1991) "The great Eskimo vocabulary hoax."
[8] For a refined version using schematic proplets see 2.6.1.

The agents are concrete individuals with real bodies[9] existing in the real world. The external surface of the word water is shown as a letter sequence between the agents (in a graph, a surface is shown easier as writing than as speech).

Such an external surface exists in a certain modality as a concrete external object with a particular shape, which may be measured and described with the methods of the natural sciences. Such an external surface has neither a content attached to it nor any grammatical properties. Its shape is arbitrary in the sense that it does not matter whether the agents use the surface water or the surface eau as long as they obey the conventions of their common natural language.

In their minds, in contrast, both agents contain the word water consisting of the content, the surface, and the convention-based connection between the two (2.1.1). As purely cognitive representations (e.g. binary code), the analyzed words are independent of any modality,[10] but include the contents (meanings) and the grammatical properties.

The speaker realizes the external surface by copying the internal word form surface and performing it externally in the modality of choice (*copy-realize*). The hearer recognizes the external surface by matching it with a learned surface and by using the recognized surface for lexical lookup (*match-lookup).*[11]

Even though the internal representations of surfaces and contents are modality-free[12] in principle, they may relate indirectly to the modality of the external interface by which they have been recognized or realized. During recognition an internal surface matches a modality-dependent representation such as sound, and during production it may be realized in such a modality.

In this indirect sense, the internal surfaces are usually mono-modal[13] while the contents are multi-modal. For example, the surface of the word blueberry is mono-modal in that the associated modality is purely acoustic, optical, or tactile (Braille), but does not include taste or color. The content of this word, in contrast, may be viewed as relating to a multi-modal conglomerate.

---

[9] The importance of agents with a real body (instead of virtual agents) has been emphasized by Emergentism (MacWhinney 1999, 2008).

[10] To illustrate the agent-internal representation of the word water in 2.2.1, we had to resort to a suitable modality, namely vision. Though unavoidable, this is paradoxical insofar as the words in the cognition of an agent are inherently modality-free, like the numbers in mathematics.

[11] One may recognize a surface as a real word, yet not know its content, for example “reprobate,” “exigency,” “prevaricate,” “crepuscular,” or “effloresce” in English. In such a case, there is an entry missing in the hearer’s lexicon. This may hold even more when recognizing a surface-like shape as a word form of a foreign language.

[12] Barsalou (1999) calls modality-free *modality-general* and modality-dependent *modality-specific*. It seems that a number in mathematics, for example, is modality-free rather than modality-general.

[13] A multi-modal representation of language surfaces may be found in an opera performance in which the text sung on stage (acoustic modality) is also displayed more or less simultaneously in writing above or below the stage (optical modality). A more mundane example is an English movie with English subtitles.

The reliance of the natural language information transfer mechanism on modality-dependent unanalyzed external surfaces is formulated as the First Mechanism of Communication:[14]

2.2.2 First Mechanism of Communication (MoC-1)

> Natural language communication is based on ordered modality-dependent unanalyzed external surfaces which have neither content (meaning$_1$) nor any grammatical property.

Given that the external surfaces are modality-dependent, while their internal, cognitive counterparts are modality-free, there is a constant mapping between modality-dependent and modality-free representations during communication (1.2.3, Conversion Universal 2). More specifically, in the speak mode, the mapping is from cognitive, agent-internal, analyzed, modality-free surfaces to noncognitive, agent-external, unanalyzed, modality-dependent surfaces. In the hear mode, the mapping is from noncognitive, agent-external, unanalyzed, modality-dependent surfaces to cognitive, agent-internal, analyzed, modality-free surface representations.

It follows from MoC-1 that reconstructing the mechanism of natural language communication cannot be limited to a grammatical analysis of isolated language signs, but must include the processing of the communicating agents:

2.2.3 Functional model of natural language communication

A functional model of natural language communication requires

1. cognitive agents each with (i) a body, (ii) external interfaces for recognition and action, and (iii) a memory for the analysis, storage, and processing of content,
2. external language surfaces which can be recognized and produced by the external interfaces of these agents using pattern matching,
3. agent-internal (cognitive) surface-content pairs established by convention and stored in the memory of the agents, whereby the shapes of the internal surfaces correspond to those of the external ones, and

[14] MoC-1 differs from the [-constructive] ontology (FoCL Sect. 20.4) of mathematical realism,, e.g. Montague grammar (Montague 1974), Situation Semantics (Barwise and Perry 1983), and Discourse Semantics (Kamp 1980). In [-constructive] systems, the relation between the language signs and an artificial "world" is established directly in the form of truth-conditions defined by logicians in a metalanguage as relations between surfaces and abstract (set-theoretic) objects.

4. an agent-internal algorithm which constructs complex contents from elementary ones by establishing semantic relations between them.

In short, the cognition of the agents in a language community is as important for modeling natural language communication as are the language signs.

This may be shown by "lost languages" (FoCL Sect. 5.3). Imagine the discovery of clay tablets left behind by an unknown people perished long ago. Even though the signs of their language are still present in the form of glyphs on the tablets, the language as a means of communicating meaningful content is lost. The only way to revive the language at least in part is to reconstruct the *knowledge* of the original speakers – which was part of their cognition and is now part of the cognition of the scientists.[15]

The requirements of 2.2.3 are minimal. They are sufficient, however, to distinguish natural language from other forms of communication:

2.2.4 Forms of communication without natural language

- endocrinic messaging by means of hormones,
- exocrinic messaging by means of pheromones, for example, in ants,[16] and
- the use of samples, for example, in bees communicating a source of pollen.

These forms of communication differ from natural language because they lack a set of external surfaces with corresponding internal surface-content pairs established by convention.[17]

From a functional point of view, the mechanism described by MoC-1 has the following advantages for communication:

2.2.5 Advantages following from MoC-1

1. Modality-free internal contents (meanings) attached to internal surfaces are not restricted by the modality of the external surfaces.
2. Modality-dependent external surfaces are better suited for (i) content transfer between agents and (ii) agent-external long-term storage than the associated agent-internal modality-free contents.

[15] Thereby, knowledge of the word form contents and the algorithm for their composition alone is not sufficient for a complete interpretation. What is needed in addition is the archaeologists' educated conjecture about the original writer's context of interpretation.

[16] Wilson (1998, p. 229) describes the body of an ant worker as "a walking battery of exocrinic glands."

[17] Another example of not constituting a language is the macros used in programming languages. Defined ad hoc by the programmer as names for pieces of code, they are (program-)internal abbreviations. As such they do not require any external surfaces agreed on by convention and are not used for inter-agent communication. Nevertheless, as abbreviation devices, macros realize an important technique which is also available in natural language.

The advantage of (1) may be illustrated by a sign-theoretic comparison of symbols and icons (FoCL Sect. 6.4). Icons in the visual modality, for example, are limited to a visual representation of content, which makes it often impossible to represent concepts from another modality, for example, the content of *sweet*. Symbols, in contrast, have no such limitation because their content representations (meaning$_1$) are modality-free.[18]

The advantage of (2) is that the transfer and storage of an unanalyzed external surface is simple and robust compared to the often delicate meanings, i.e. contents attached by convention to surfaces within the agents' cognition. This holds especially for the written representation of language, which has long been the medium of choice for the agent-external storage of content.[19]

Naturally, the specialized and powerful technique characterized by MoC-1 also has an apparent disadvantage: because the surface for a content can take any shape within the limits of its modality, communities of natural agents can and do evolve their own languages. This results in the difficulty of communicating in foreign language environments.

2.3 Modality Conversion in the Speak and Hear Modes

MoC-1 is not only a conceptual insight into the working of natural language communication between agents viewed from the outside, but constitutes a well-defined technical challenge, namely the construction of machines which can recognize and produce external language surfaces. The basic task of these machines may be viewed as an automatic conversion between a modality-dependent realization of an agent-external surface and its modality-free counterpart represented as agent-internal digital code, e.g. seven-bit ASCII. This conversion is instantiated in four basic variants, namely the speak mode and the hear mode each in the modalities of vision and audition.

In the hear mode, today's systems of *speech recognition* convert external acoustic surfaces into modality-free digital code, mainly by the statistical method of Hidden Markov Models (HMMs). In the optical modality, today's

[18] A modality-free (declarative) definition of a basic content is procedural in the sense that it is distilled from the recognition and action procedures of the cognitive agent. Whether such a content representation in an artificial agent is adequate or not is decided by the agent's performance. For example, an artificial agent's concept of *shoe* may be considered adequate if the agent picks out the same object(s) from a collection of different things as a human would (Sects. 8.5, 15.6; NLC Sect. 4.3).

[19] Recently, storage by means of written language has been complemented by the technologies of tape recording, video, CD, DVD, etc., which allow to preserve also spoken and signed language for indefinite periods of time.

On the relation between modalities and media see FoCL Sect. 1.4 and NLC Sect. 2.2.

systems turn images of letters into modality-free digital code based on the software of *optical character recognition* (OCR).

In the speak mode, today's systems of *speech synthesis* convert digitally represented text into artificially realized speech, basically by concatenating pieces of recorded speech. In the modality of vision, there is the conversion from digital code to the familiar letter images on our computer screens, which may be called *optical character synthesis* (OCS).

Signs are interpreted relative to their spatio-temporal *origin* (Sect. 11.1; FoCL 5.3.4). For the speaker, this origin is the production of the sign's surface, regardless of the modality. For the hearer, in contrast, there is an important difference between the interpretation of spontaneous language in face-to-face communication and of recorded language.

In face-to-face dialog, the hearer witnesses the origin of the surface directly, but in recorded language the moment and place of interpretation may be far removed from the spatio-temporal location of the sign's production. Therefore a writer must encode the coordinates of Space, Time, Agent, and intended Recipient (STAR) of a sign's production into the sign, and a reader must reconstruct the sign's utterance meaning$_2$ relative to these values.

The visual and the auditory modality differ also in that recorded language may be corrected by the author, while spontaneous speech can not:

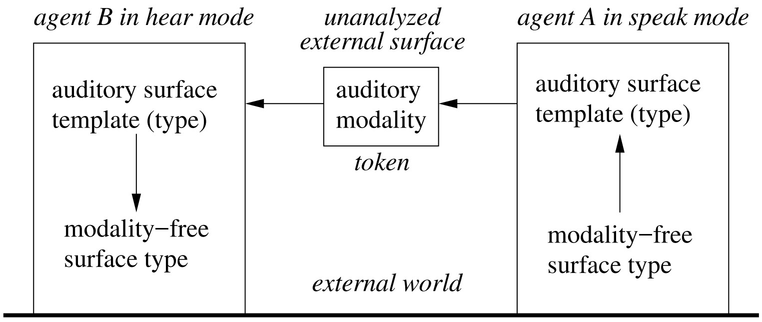
> Speech is irreversible. That is its fatality. What has been said cannot be unsaid, except by adding to it: to correct here is, oddly enough, to continue.
>
> R. Barthes (1986, p. 76)

Of course, as soon as spoken language is recorded, an interpretation may be arbitrarily distant in time and space from its production. Also, a recording may be "doctored" – which from a certain point of view is a form of correction.

In any modality, however, the hear mode (token-type mapping from a modality-dependent to a modality-free representation, 1.2.3) abstracts away from the properties of a external surface such as speed, pitch, intonation, etc. in spoken language, and font, size, color, etc. in written language, which from a certain point of view may be regarded as accidental. The converse holds in the speak mode (type-token mapping from a modality-free to a modality-dependent representation), which must settle on such properties as pitch, speed, dialect, etc., for the realization of the external surface.

The transitions between a modality-free surface type and a modality-dependent external surface token require a modality-dependent surface template (type) as an intermediate step, illustrated below with the auditory modality:

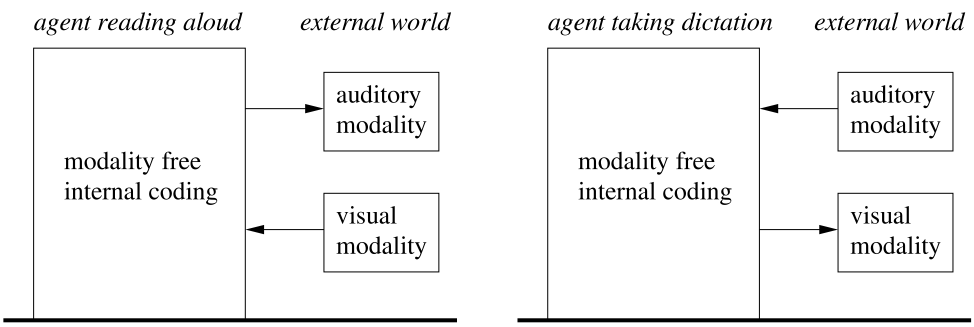
2.3.1 MEDIATION BY MEANS OF MODALITY-DEPENDENT TEMPLATE



In the speak mode, the template mediates between the modality-free internal surface type and the modality-dependent external surface token by selecting the modality-dependent properties, just as it used in the hear mode to remove them. For each modality, the agent uses the associated modality-dependent surface templates.

If the speak mode and hear mode utilize the *same* modality and are realized by *different* agents, as in example 2.3.1, we have inter-agent communication. If the speak mode and the hear mode utilize *different* modalities and are realized by the *same* agent, in contrast, we have a modality conversion. In nature, modality conversion is illustrated by reading aloud (conversion from the visual to the auditory modality) and taking dictation (conversion from the auditory to the visual modality).

2.3.2 TWO KINDS OF MODALITY CONVERSION



In technology, reading aloud is modeled by combining optical character recognition and speech synthesis, which is an important application for the blind. Conversely, taking dictation is modeled by a machine called "electronic secretary," which combines speech recognition with optical character synthesis.[20]

[20] Infrared cameras are another technical means of modality conversion, representing temperature (what has been called the *temperature modality* by Dodt and Zotterman 1952) as color (visual modality).

2.4 General-Purpose Interfaces and the Language Channels

In recognition, language interpretation is embedded into nonlanguage recognition. For example, in the visual modality humans can recognize nonlanguage[21] input such as streets, houses, cars, trees, and other agents as well as language input such as shop signs and text in newspapers, books, and letters. Similarly in the auditory modality: humans can recognize nonlanguage input such as the sounds made by rain, wind, birds, or approaching cars, as well as language input such as speech from other agents, radio, and television.

In action, language production is likewise embedded into nonlanguage action. For example, in the modality of manipulation humans can use their hands for generating nonlanguage output, such as filling the dishwasher or drawing a picture, as well as for language production, such as hand-writing or typing on a computer keyboard. Similarly in the modality of articulation: humans can use their voice for nonlanguage output, such as for singing without words or for making noises, as well as for language production, i.e. for speech.

Such embedding of language recognition and synthesis into nonlanguage recognition and action cannot be handled by today's technologies of speech recognition and optical character recognition. Instead, they use input and output channels dedicated to language. This constitutes a substantial simplification of their respective tasks (smart solution).[22]

Despite this simplification, automatic speech recognition has not yet reached its goal.[23] For proof, there is no need to argue; we simply point to the ever-increasing number of keyboards in everyday use. If automatic speech recognition worked in any way comparable to the speech recognition capability of an average human,[24] few users would prefer the keyboard and screen over speech.

For building a talking robot like C3PO any time soon, insufficient automatic speech recognition presents an – almost certainly temporary – obstacle. Even now, however, it does not hinder the current work on the computational reconstruction of natural language communication in DBS. The reason is an important difference between natural and artificial cognitive agents regarding what we call the *auto-channel* and the *service channel* (NLC 1.4.3).

[21] We prefer the term nonlanguage over nonverbal because the latter leads to confusion regarding the distinction between verbs, nouns, and adjectives.

[22] For the distinction between smart and solid solutions see FoCL Sect. 2.3

[23] Optical character recognition, in contrast, is quite successful, especially for printed language, and widely used for making paper documents available online.

[24] Technically speaking, a practical system of speech recognition must fulfill the following requirements simultaneously (Zue et al. 1995):
1. speaker independence,
2. continuous speech,
3. domain independence,

The only channel a natural agent has for recognition and action is the auto-channel. It evolves naturally during child development and comprises everything a natural agent can see, hear, feel, taste, etc., as well as consciously do. The auto-channel also includes natural speech recognition and production.

In a standard computer, e.g. a notebook or a desktop computer, in contrast, there is no auto-channel. Instead, there is only a service channel consisting mainly of the keyboard and the screen. It allows users and experts alike to access the hardware and software of the computer directly.

Building a talking robot consists largely in reconstructing an artificial auto-channel, with external interfaces for recognition (vision, audio) and action (articulation, locomotion, manipulation). For maintaining the agent in a continuous state of balance, the auto-channel must be connected to the agent's autonomous control (Chap. 5; Sect. 14.3).

The computational reconstruction of the auto-channel is an incremental process which relies heavily on the keyboard and the screen as the service channel for direct access to, and control of, the hardware and software of the robot under development. After completion, the artificial agent will be able to interact autonomously with the external world via its auto-channel, including communication with the human user. However, in contrast to a natural agent, an artificial agent will not only have an auto-channel, but also a service channel as a remnant of its construction process.

The essential role of the service channel for bootstrapping the reconstruction of cognition is especially clear in natural language interpretation and production. This is because it does not really matter for upscaling whether a surface gets into and out of the computer via the auto-channel or via the service channel. What matters is that there is a transfer of word form surfaces between the scientist and the computer at all, and for this the keyboard and the screen of today's standard computers are sufficient.

The possibility of reconstructing natural language processing in artificial agents via the service channel is good news for developing capable automatic speech recognition.[25] The reason for a 20 year stagnation in this area is a search space too large for today's statistical approaches. The gigantic search space is caused by the large number of possible word forms in a natural language

4. realistic vocabulary,
5. robustness.

No system of speech recognition has yet achieved these desiderata simultaneously.

[25] Contextual cognition, such as nonlanguage vision and audio, may also benefit from the service channel. By building a context component with a data structure, an algorithm, and a database schema via direct access, the robot's auto-channel is provided with a component to map into and out of.

multiplied with an even larger number of possible syntactic combinations and variations of pronunciation between different speakers.

The best way to reduce this search space is by hypotheses on possible continuations computed by the time-linear LA grammar algorithm, enriched with statistical information distilled from an RMD corpus (Sects. 15.3–15.6) as well as expectations based on domain and general world knowledge.[26] After all, this is also the method used by humans for disambiguating speech in noisy environments.[27] Using this method for artificial speech recognition requires a functional theory of how communicating with natural language works.

2.5 Automatic Word Form Recognition

Assuming that the language input to an artificial cognitive agent is word form surfaces provided by the service channel as sequences of letters, the first step of any rule-based (i.e. not purely statistical) reconstruction of natural language understanding is building a system of automatic word form recognition.[28] This is necessary because for the computer a word form surface like learns is merely a sequence of letters coded in seven-bit ASCII, no different from the inverse letter sequence snrael, for example (15.1.1, level-two).

Automatic word form recognition takes an unanalyzed surface, e.g. a letter sequence like learns, as input and provides the computer with the linguistic information needed for syntactic-semantic processing. Any such system must provide (i) *real time performance*, (ii) *categorization* and (iii) *lemmatization* (FoCL Sect. 13.4). Categorization specifies the grammatical properties, which in the case of learns would be something like "verb, third person singular, present tense." Lemmatization specifies the base form, here learn, which is used to look up the content (meaning$_1$) common to all the word forms of the paradigm, i.e. learn, learns, learned, and learning.

The recognition algorithm in its most primitive form consists of matching the surface of the unknown letter sequence with the corresponding surface (key) in a full-form lexicon,[29] thus providing access to the relevant lexical information:

[26] For example, world knowledge disambiguates the homophones The mother and her week-old calf (from a NatGeo Wild tv program) vs. The mother and her weak old calf. The second reading may be excluded by world knowledge because a calf is young. Excluding the second reading by probability alone is not feasible because even in big data the number of examples would be too small. Yet another reading is calf as part of the leg.

[27] Juravsky and Martin (2009, p. 16–17) sketch an approach which could be construed as being similar to DBS. Unfortunately, their system is neither agent-based, nor surface compositional, nor time-linear.

[28] We begin with the hear mode as a means to get content into the agent's memory (database). The availability of such content is a precondition for selective activation and inferencing as the basis for building the speak mode.

2.5.1 Matching an unanalyzed surface onto a key

unanalyzed word form surface:	learns
	matching
morphosyntactic analysis:	[learn/s, categorization, lemmatization]

In DBS, automatic word form recognition is based on the allomorph method. It combines low space requirements and optimal time efficiency with the ability to recognize neologisms. The algorithm consists of (i) segmenting the letter sequence of a surface into known but unanalyzed parts, called allomorphs, (ii) lexical lookup of the corresponding analyzed allomorphs in a trie structure,[30] and (iii) their composition into well-formed analyzed word forms (FoCL Chap. 14). This requires (i) an online lexicon for base forms (morphemes), e.g. wolf, (ii) allo-rules for deriving variants (allomorphs) of the base form, e.g. wolf and wolv-, before runtime, and (iii) combi-rules for (a) combining the analyzed allomorphs during runtime, e.g. wolv/es, and (b) deriving categorization and lemmatization for the complete word form.

Building such a system of automatic word form recognition for any given natural language is not particularly difficult, even for writing systems based on characters,[31] e.g. Chinese and Japanese, rather than letters. Given (i) an online dictionary of the natural language of choice, (ii) a suitable off-the-shelf software framework, and (iii) a properly trained computational linguist, an initial system can be completed in less than six months.[32] It will provide accurate, highly detailed analyses of about 90% of the word form types in a corpus.

Increasing the recognition rate to approximately 100% is merely a matter of additional work. It consists of adding missing entries to the online lexicon, and improving the rules for allomorphy and for inflection/agglutination, derivation, and composition. To maintain a recognition rate of practically 100%, the system must be serviced continually, based on a RMD corpus (Sect. 15.3).

The improvements of an automatic word form recognition system apply to the *living language*, as a continuous stream of fresh data in spontaneous dialog. Statistical tagging (Sect. 15.2), in contrast, does not run in real time and is therefore is limited to language fixed in storage. Also, manual postediting has

[29] Full-form lookup is one of three basic methods of automatic word form recognition, the others being the morpheme method and the allomorph method. FoCL Chaps. 13–15.

[30] Briandais (1959); Fredkin (1960); Knuth (1998, p. 495–512); FoCL 14.3.3.

[31] A trie structure may be used for recognizing characters because the strokes are ordered in a sequence, just like the sequence of letters in Western writing systems. Both systems allow alternative segmentations (ambiguity), e.g. German Stau/becken vs. Staub/ecke/n (FoCL 14.3.2).

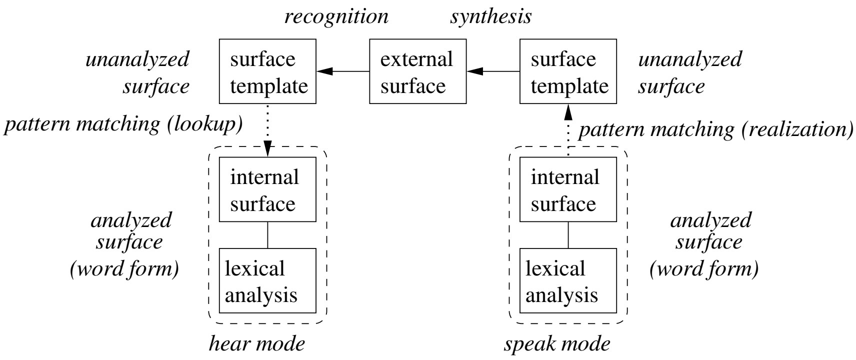
[32] This is the standard period of time for writing an MA thesis at the University of Erlangen-Nürnberg.

to be done for each corpus from scratch and may take more than a decade, as shown by the BNC (Burnard ed. 1995 and 2007).

2.6 Backbone of the Communication Cycle

The first step of natural language interpretation in the hear mode is automatic word form recognition. The last step of language production in the speak mode is automatic word form synthesis. The two mechanisms are alike in that they establish a correlation between unanalyzed and analyzed word form surfaces by means of pattern matching, but differ in the direction:

2.6.1 BACKBONE OF SURFACE-BASED INFORMATION TRANSFER



At the center of the inter-agent information transfer is the modality-dependent unanalyzed *external surface* token (MoC-1, 2.2.2).

In the speak mode, an internal surface is mapped into a template which synthesizes a corresponding external surface in a certain modality. In the hear mode, the external surface is recognized by matching it with an internal template which retrieves a corresponding analyzed word form (lexical lookup). The surface templates are types which are modality-dependent, while the internal surfaces are types which are modality-free (2.3.1).

In successful communication, the unanalyzed external surface must correspond to the analyzed surface (word form) in the speaker and the hearer. This requires that they associate the internal surface with the same lexical analysis, i.e. they must speak the same language. In addition, the word forms are assumed to match an agent-internal context of interpretation (4.3.2, 4.3.3).

The physiological precondition for language communication is the speaker having the synthesis and the hearer having the recognition interface for the modality of the surface being used. An agent without external interfaces for the speak and the hear mode cannot communicate with language at all. If agent

A tries to communicate in the modality of speech and agent B is deaf, the attempt will fail, and similarly if agent B tries to communicate in writing and agent A is blind. In short, a model of natural language communication requires that agents have interfaces (i) in principle and (ii) in the right modality.

This has been preformulated in NLC Sect. 1.5, as the following principle:

2.6.2 Principle of Interface Compatibility (IC)

> For unrestricted human-machine communication, the artificial cognitive agent with language (DBS robot) must be equipped with external interfaces which are compatible with those used by the human prototype for communication.

The external interfaces of humans are concretely given and therefore suitable for an objective structural and functional analysis: using a combination of the natural sciences and engineering, it should not be too difficult to fit the artificial agent with interfaces which are functionally compatible with those of the natural prototype.

An equivalence principle, also preformulated in NLC Sect. 1.5, relates the processing order of the hear mode (input) and the speak mode (output) to the one-dimensional, linear structure of the language signs:

2.6.3 Equivalence Principle for Input/Output (IO)

> In natural language interpretation, the artificial agent must analyze the modality-dependent external surface by
> (i) segmenting it in the same way into parts (i.e. word forms) and
> (ii) ordering the parts in the same way (i.e. in a time-linear sequence)
> as the human prototype; and accordingly for language production.[33]

For the artificial agent, IO equivalence extends the backbone of surface-based information transfer (2.6.1) from single word forms to complex expressions.

[33] In a known written language, the computational segmentation of a word form into allomorphs and, if necessary, a sentence into word forms (continua) is handled by means of a trie structure, a suitable online lexicon, and an algorithm of automatic word form recognition (FoCL Sect. 14.3). For continuous speech in an unknown language, in contrast, determining the correct segmentation requires the help of a native speaker. As pointed out by Sapir (1921, p. 33), aborigines who do not read and write can nevertheless dictate in their language word form by word form.

To establish the correct segmentation in a sentence "scientifically," even in a known language like English, Harris (1951) used the method of movement and substitution tests (FoCL 8.4.7, 8.4.8). These were turned by his student Chomsky into the transformations of nativism and declared innate. By using the partial derivation order of possible *substitutions* instead of computing possible *continuations*, nativism and American structuralism have consistently disregarded the structural relation between the segmentation and the time-linear order of the surface as the backbone of semantic-syntactic encoding (speak mode) and syntactic-semantic decoding (hear mode) of natural language.

The linear order of the word forms in a surface constitutes the unique *mode of composition*. Together with the lexical analysis of the word forms, this order determines[34] the compositional semantic relations.

Because a time-linear order must be provided for processing the input and output of the language surfaces anyway, the most direct means of connecting the speak and the hear mode is a time-linear derivation order also for the think mode, by which they are connected. In this way, a constant switching between computing possible continuations and possible substitutions is avoided,[35] thus providing a smooth functional flow from input to output.

Does computing possible continuations for the full cycle of natural language communication work empirically? So far, the DBS analysis of different grammatical constructions, such as clausal subject (NLC Sect. 7.1), clausal object (NLC Sect. 7.2), relative clauses (NLC Sects. 7.3, 7.4; Sect. 9.3), adverbial clause (NLC Sect. 7.5), subclause recursion (NLC Sect. 7.6; Sect. 9.4), gapping (NLC Sects. 8.5, 8.6, Chap. 10), infinitive (Sect. 15.4), bare infinitive (15.6.2), active/passive (9.2.1, 9.2.2), copula constructions (NLC 6.6.9–6.6.11), etc., in many different languages (3.5.2) has not discovered any empirical difficulty (fatal flaw) for the time-linear syntactic-semantic interpretation and semantic-syntactic production of natural language.

This is different for phrase structure grammar and truth-conditional semantics. Apart from failing to be interface compatible (IC 2.6.2) and input/output equivalent (IO 2.6.3), they each have a characteristic fatal flaw regarding the empirical analysis of natural language on their own sign-based terms.

More specifically, context-free phrase structure grammar cannot handle discontinuous elements, as shown by Bar-Hillel (1953).[36] Truth-conditional semantics cannot allow the words true and false in the object language (which is what the talking robot would be using), as shown by Tarski (1935).[37]

In DBS, discontinuous elements do not exist because the linguistic analysis is based on a time-linear build-up and canceling of valency positions, and not on the direct and exhaustive dominance of constituent structure (12.2.1). Also, there is no need to forbid the words true and false in the artificial agent's vocabulary: the Epimenides paradox is naturally absent in DBS because basic contents (meanings) are defined as types of the agent's recognition and action procedures (grounding), and not as truth conditions.

[34] Fregean Principle (1.4.2). Compare The dog bit the man and The man bit the dog. The two expressions consist of the same word forms, but differ in order, resulting in different semantic relations.

[35] Except for subintelligent agents who switch from the hear mode directly into the speak mode, i.e. without an intervening think mode.

[36] Constituent structure paradox, Sect. 12.2; FoCL Chap. 8.

[37] Epimenides paradox, FoCL Sect. 19.5; Betti (2004).

3. Mystery Number Two: **Natural Language Communication Cycle**

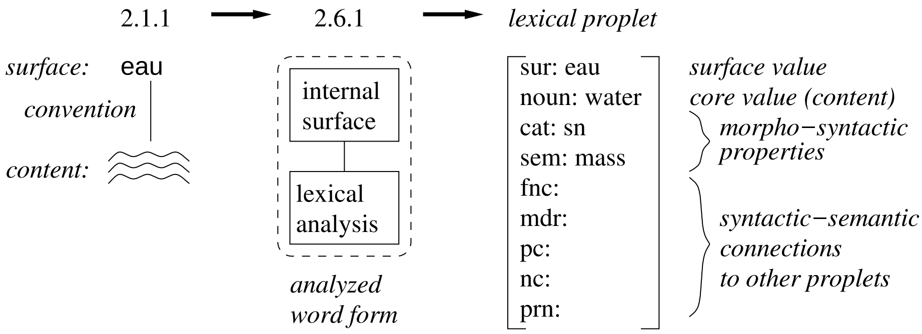
The second mystery is the nature of *content* in such functions as interpreting and producing language surfaces, storing in and retrieving from memory, and inferencing. While the surface of a natural language expression is given concretely as a modality-dependent token, the associated encoding and processing of content must be reconstructed via functional equivalence (FE, 1.1.1).

3.1 Data Structure

From a software engineering point of view, choosing the format of the basic units of a system amounts to specifying its data structure.[1] A well-defined data structure is required for designing the input and output the system's external interfaces, for storage and retrieval, and for the algorithm which operates on the data. A data structure is like the concept of a cell in biology.

In the hear mode of a talking robot, the segmentation of the incoming stream of unanalyzed raw language data is linguistically correct only if the result is a sequence of word form surfaces. Therefore, word forms are the natural candidates for being analyzed as the basic data structure. The format of *proplets* (1.5.1) as the data structure of DBS has evolved from earlier stages:

3.1.1 DEVELOPMENT OF THE PROPLET FORMAT AS A DATA STRUCTURE



[1] In early computer science, the term "data structure" referred to hardware constructs such as queue, stack, linked list, car, cdr, array, or pointer. The development of higher programming languages

Representation 2.1.1, inspired by de Saussure, is made more specific in 2.6.1 by stipulating that a surface-content pair is agent-internal, and made more general by using lexical analysis to represent (i) the semantic core and (ii) the morphosyntactic properties. The formats 2.1.1 and 2.6.1 are alike in that they use a graphical means (vertical line) to indicate the conventional connection between the surface and the lexical properties.

The *lexical proplet* in 3.1.1 differentiates the analysis further by using the format of non-recursive feature structures with ordered[2] attribute-value pairs (avps). This simple format is (i) suitable for storage and retrieval in a database, and allows the coding of (ii) any morphosyntactic properties, (iii) any inter-proplet relations using addresses, and (iv) the convention-based connection between the internal surface and the lexical analysis without graphical means.[3]

In a proplet, the lexical and the compositional aspects of a word form are systematically distinguished by means of characteristic attributes.[4] For example, the lexical aspect in the proplet shown in 3.1.1 is represented by the values of the first four attributes: 1. [sur: eau] specifies the *surface*; 2. [noun: water] consists of the core *attribute* noun and the core *value* water specifying the literal meaning represented by a concept name; 3. [cat: sn] (for *category: singular noun*) specifies the morphosyntactic properties contributing directly to the combinatorics, for example, regarding agreement; 4. [sem: mass] (for *semantics: mass*[5]) specifies morphosyntactic properties not directly contributing to the combinatorics.

brought into use the term “abstract data type” to refer to “software” data structures, which are specified independently of the hardware base. With the increasing abstracting away from the hardware level, the two terms are now widely used interchangeably. This makes it possible to call proplets, specified as non-recursive feature structures with ordered attributes, a “data structure,” which is shorter than “abstract data type” and makes better use of the earlier term.

[2] According to ISO 24610-1 (TC37/SC4), a standard feature structure is a(n unordered) *set* of attribute-value pairs (Carpenter 1992), presumably for reasons of misguided mathematical aesthetics. Our alternative of defining proplets as a(n ordered) *list* of attribute-value pairs has the advantages of easier readability and a more efficient computational implementation, e.g. for pattern matching (1.5.3).

[3] A precursor of proplets, used in NEWCAT and in FoCL Chaps. 16–18, is a format of ordered triples, consisting of (i) the surface, (ii) the category, and (iii) the base form. For example, the word form gave was analyzed as the triple [gave (n′ d′ a′ v) give], which may be transformed into a character-separated value (csv) notation, as used in the original WordNet (Miller1996), CELEX2 (Olac Record 1996), and other lexical encodings. The corresponding proplet notation would be

$$\begin{bmatrix}\text{sur: gave}\\ \text{verb: give}\\ \text{cat: n}'\ \text{d}'\ \text{a}'\ \text{v}\\ \ldots\end{bmatrix}$$
, which is a flat (non-recursive) feature structure with ordered attributes.

[4] This is in contradistinction to statistically-based tagging, in which the lexical analysis of a word form and its syntactic-semantic role in a sentence are inextricably fused. As a result, the number of CLAWS4 tagged word forms in the BNC is 37.5% greater than the number of unanalyzed surfaces (FoCL Sect. 15.5.), constituting a concealed form of lexical ambiguity. See also Sect. 15.2.

[5] A mass noun is in contradistinction to a count noun, which refers to a discrete object.

The compositional aspect of the word form is represented by the remaining features. Because the proplet illustrated in 3.1.1 is lexical, the compositional attributes have no values yet, but will receive them during syntactic-semantic parsing in the hear mode (3.3.1). The *continuation* attributes fnc (functor) and mdr (modifier) will characterize the relations of the proplet to other proplets in terms of functor-argument, while pc (previous conjunct) and nc (next conjunct) will characterize relations of coordination. The value of the *bookkeeping* attribute prn (proposition number), finally, will be a number common to all proplets belonging to the same elementary proposition and indicate the temporal order of arrival in the database relative to other propositions.

3.2 Three Requirements on Proplets as a Data Structure

The first requirement on proplets as a data structure is that they provide for a simple, concrete, and precise representation of propositional content. As an example of an elementary intrapropositional functor-argument, consider the content of Julia knows John., represented as a set of linked proplets:

3.2.1 FUNCTOR-ARGUMENT OF Julia knows John.

$$\begin{bmatrix}\text{noun: Julia}\\ \text{cat: snp}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix} \begin{bmatrix}\text{verb: know}\\ \text{cat: \#s3}' \text{ \#a}' \text{ decl}\\ \text{arg: Julia John}\\ \text{prn: 625}\end{bmatrix} \begin{bmatrix}\text{noun: John}\\ \text{cat: snp}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix}$$

The simplified proplets are held together by a common prn value (here 625). The functor-argument[6] is coded solely in terms of attribute values. For example, the *Julia* and *John* proplets specify their functor as know, while the *know* proplet specifies Julia and John as its arguments.[7]

The second requirement is that proplets are suited for efficient computational pattern matching. Consider turning the content 3.2.1 into a maximal pattern:

3.2.2 TURNING 3.2.1 INTO A PATTERN

$$\begin{bmatrix}\text{noun: } \alpha\\ \text{fnc: } \beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta\\ \text{arg: } \alpha\ \gamma\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: } \gamma\\ \text{fnc: } \beta\\ \text{prn: K}\end{bmatrix}$$

[6] When we refer to a proplet by its core value, we use italics, e.g. *know*, whereas for reference to an attribute or a value within a proplet we use helvetica, e.g. fnc or know.

[7] Lexical name proplets are special in that they have a sur value, but no core value (3.5.3). A core value is provided nonlexically as the “named referent” by an act of generalized baptism. In the hear mode, names are accessed by their sur value, but in the speak mode by the named referent (NLC 2.6.7, PoP-6; HBTR Sect. 3.4). For simplicity, the surface of a name is used here as the noun value, like a regular core value, in the speak and the hear mode (e.g. 3.2.1, 3.3.1, 4.1.2).

The semantic relations between the pattern proplets in 3.2.2 are the same as those between the content proplets in 3.2.1. The pattern is maximal because *all* constants in 3.2.1 are replaced with variables by simultaneous substitution. A pattern matches the content which it has been derived from as well as an open number of similar contents[8] (6.5.1–6.5.4):

3.2.3 MATCHING BETWEEN PATTERN 3.2.2 AND CONTENT 3.2.1

$$
\begin{array}{lccc}
\textit{pattern level} & \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: }\beta\\ \text{arg: }\alpha\ \gamma\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{noun: }\gamma\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix}\\
\quad\textit{matching and binding} & & & \\
\textit{content level} & \begin{bmatrix}\text{noun: Julia}\\ \text{cat: snp}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix} & \begin{bmatrix}\text{verb: know}\\ \text{cat: \#s3}'\ \text{\#a}'\ \text{decl}\\ \text{arg: Julia John}\\ \text{prn: 625}\end{bmatrix} & \begin{bmatrix}\text{noun: John}\\ \text{cat: snp}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix}
\end{array}
$$

The matching between the pattern and the content is successful because the pattern proplets share[9] the same attributes in the same order as the corresponding content proplets and the distribution of variables at the pattern level corresponds to the distribution of constants at the content level.

The relations in 3.2.1–3.2.3 are *duplex* (16.6.7) in that the core value of *Julia* reappears in the arg slot of *know* and the core value of *know* reappears in the fnc slot of *Julia*, and similarly for the object-predicate relation. Duplex coding is needed by the think mode navigation for returning to the top verb (7.6.6).[10]

The exception is extrapropositional coordination (DBS.Nav in NLC 12.1.1, 14.5.1). For navigating from one proposition to the next, there is no higher level to return to, as in the following example:

3.2.4 COORDINATION STRUCTURE OF Julia sang. Sue slept. John read.

$$
\begin{bmatrix}\text{noun: Julia}\\ \text{fnc: sing}\\ \text{prn: 10}\end{bmatrix}
\begin{bmatrix}\text{verb: sing}\\ \text{arg: Julia}\\ \text{nc: (sleep 11)}\\ \text{pc:}\\ \text{prn: 10}\end{bmatrix}
\begin{bmatrix}\text{noun: Sue}\\ \text{fnc: sleep}\\ \text{prn: 11}\end{bmatrix}
\begin{bmatrix}\text{verb: sleep}\\ \text{arg: Sue}\\ \text{nc: (read 12)}\\ \text{pc:}\\ \text{prn: 11}\end{bmatrix}
\begin{bmatrix}\text{noun: John}\\ \text{fnc: read}\\ \text{prn: 12}\end{bmatrix}
\begin{bmatrix}\text{verb: read}\\ \text{arg: John}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}
$$

The propositions with the prn values 10, 11, and 12 are concatenated by the nc value only (simplex). A backward traversal in a top level coordination content (Sect. 5.5) requires the use of inferencing.

Similar to the transition from the content 3.2.1 to the pattern 3.2.2, the pro-

---

[8] A DBS pattern allows the use of detailed syntactic and semantic properties for efficient high-resolution retrieval (Sect. 5.4).

[9] For successful matching, the attributes of the pattern proplet must be (i) a *sublist* ($\leq$) of the attributes of the content proplet and (ii) corresponding variables and constants must be compatible (NLC 3.2.3).

[10] A semantic relation between proplets A and B is called *slant duplex* if (i) the (address of the) core value of A appears as a continuation value in B, and (ii) the (address of the) core value of B appears in a continuation slot of A. If only one condition is fulfilled, a relation is called *slant simplex* (16.6.7).

plets in the extrapropositional coordination 3.2.4 may be turned into a pattern by replacing constants with variables via simultaneous substitution:

3.2.5 Turning 3.2.4 into a pattern

$$
\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{verb: }\beta\\ \text{arg: }\alpha\\ \text{nc: }(\delta\text{ K+1})\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: }\gamma\\ \text{fnc: }\delta\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{verb: }\delta\\ \text{arg: }\gamma\\ \text{nc: }(\psi\text{ K+2})\\ \text{pc:}\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{noun: }\phi\\ \text{fnc: }\psi\\ \text{prn: K+2}\end{bmatrix}
\begin{bmatrix}\text{verb: }\psi\\ \text{arg: }\phi\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K+2}\end{bmatrix}
$$

The pattern matches the content 3.2.4 from which it was derived as well as an open number of similar contents:

3.2.6 Matching extrapropositional coordination pattern

$$
\begin{array}{l}
\textit{pattern level}\quad
\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{verb: }\beta\\ \text{arg: }\alpha\\ \text{nc: }(\delta\text{ K+1})\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: }\gamma\\ \text{fnc: }\delta\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{verb: }\delta\\ \text{arg: }\gamma\\ \text{nc: }(\psi\text{ K+2})\\ \text{pc:}\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{noun: }\phi\\ \text{fnc: }\psi\\ \text{prn: K+2}\end{bmatrix}
\begin{bmatrix}\text{verb: }\psi\\ \text{arg: }\phi\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K+2}\end{bmatrix}\\
\textit{matching and binding}\\
\textit{content level}\quad
\begin{bmatrix}\text{noun: Julia}\\ \text{fnc: sing}\\ \text{prn: 10}\end{bmatrix}
\begin{bmatrix}\text{verb: sing}\\ \text{arg: Julia}\\ \text{nc: (sleep 11)}\\ \text{prn: 10}\end{bmatrix}
\begin{bmatrix}\text{noun: Sue}\\ \text{fnc: sleep}\\ \text{prn: 11}\end{bmatrix}
\begin{bmatrix}\text{verb: sleep}\\ \text{arg: Sue}\\ \text{nc: (read 12)}\\ \text{prn: 11}\end{bmatrix}
\begin{bmatrix}\text{noun: John}\\ \text{fnc: read}\\ \text{prn: 12}\end{bmatrix}
\begin{bmatrix}\text{verb: read}\\ \text{arg: John}\\ \text{nc:}\\ \text{prn: 12}\end{bmatrix}
\end{array}
$$

The computational simplicity of matching the pattern and the content level is a direct result of defining the data structure of proplets as non-recursive feature structures with ordered attributes.

The third requirement on connected proplets is that they must be *order-free*[11] in the sense that the syntactic-semantic relations holding between them are maintained regardless of how they are stored in and retrieved from the database. The order-free nature of proplets is used in the following functions:

3.2.7 Functions based on the order-free nature of proplets

1. Hear mode: the storage of proplets in the content-addressable database of a word bank (4.1.1) is determined solely (i) by the alphabetical order induced by the core value and (ii) by the moment of arrival.
2. Think mode: proplets are selectively activated by navigating along the semantic relations of structure between them, whereby the location of the next proplet is specified solely by address (primary key), independently of the kind of the semantic relation (3.2.8).

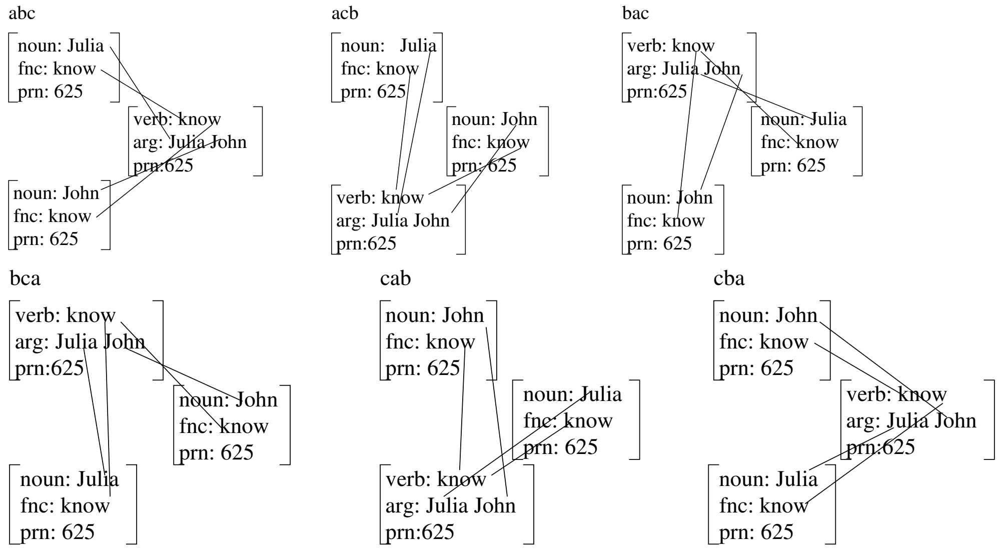
[11] This is in contradistinction to a logical formula such as $\exists x[\text{man}(x)\wedge\text{walk}(x)]$, a phrase structure tree such as 7.1.2, or a recursive feature structure with unification as shown in NLC 3.4.3, all of which may change either their meaning or lose their well-formedness if the order of their parts is changed.

3. Speak mode: the time-linear macro word order is (re)introduced by the selective activation in the think mode (3.3.3, 3.4.3).
4. Query answering: the semantic relations holding between proplets may not interfere with the retrieval of content by means of pattern matching (4.2.2).

The DBS method for making proplets order-free is coding the relations between them by *address* rather than by *embedding* (as in recursive feature structures). The mappings (i) from ordered lexical proplets to unordered content proplets in the hear mode and (ii) from unordered content proplets to ordered lexical proplets in the speak mode realize the Conversion Universal 3 in 1.2.3.

Consider how the three proplets in 3.2.1 may be stored in the orders abc, acb, bac, bca, cab, and cba without affecting their semantic relations:

3.2.8 MAINTAINING SEMANTIC RELATIONS REGARDLESS OF ORDER



The six possible proplet orders are arranged vertically. The duplex semantic relations within each triple are indicated by lines which are like rubber bands adjusting to the varying arrangements and distances (storage locations) of the proplets.[12] The addresses which connect proplets are defined by a continuation and a prn value (primary key). The nature of the semantic relation established by an address is coded by its continuation *attribute*.

3.3 Hear, Think, and Speak Modes

The fundamental principle of natural language is its time-linear structure:[13] the proper interpretation of a text is one sentence after another, of a sentence one word form after another, and of a word form one allomorph after another (FoCL 16.1.1). The speaker may decide in the middle of a sentence on how to continue (depending on the facial expression of the partner in discourse, for example). The hearer's interpretation starts with the first word form and continues as the following word forms are coming in.

In DBS, the time-linear structure is modeled by the algorithm of Left-Associative grammar. LA grammar computes *possible continuations* by always combining a current 'sentence start' with a current 'next word' into a 'new sentence start.' This amounts to interpreting a sequence like a+b+c+d... left-associatively as ...(((a b) c) d)... (Aho and Ullman 1977, p. 47).

As a simple linguistic example, consider the time-linear surface-compositional hear mode derivation of Julia knows John. The basic units, i.e. the word forms, are represented as proplets, but the operations are shown graphically by means of arrows. The result is the order-free set of proplets 3.2.1, connected into a content by means of (i) addresses and (ii) a common prn value:

3.3.1 DBS HEAR MODE DERIVATION OF Julia knows John.

	Julia	knows	John	.	
lexical lookup	noun: Julia cat: snp fnc: prn:	verb: know cat: s3' a' v arg: prn:	noun: John cat: snp fnc: prn:	verb: . cat: v' decl arg: prn:	
syntactic−semantic parsing					
1 N×FV	noun: Julia cat: snp fnc: prn: 625	verb: know cat: s3' a' v arg: prn:			*cross−copying*
2 FV×N	noun: Julia cat: snp fnc: know prn: 625	verb: know cat: #s3' a' v arg: Julia prn: 625	noun: John cat: snp fnc: prn:		*cross−copying*
3 S∪IP	noun: Julia cat: snp fnc: know prn: 625	verb: know cat: #s3' #a' v arg: Julia John prn: 625	noun: John cat: snp fnc: know prn: 625	verb: . cat: v' decl arg: prn:	*absorption*
result of syntactic−semantic parsing	noun: Julia cat: snp fnc: know prn: 625	verb: know cat: #s3 #a' decl arg: Julia John prn: 625	noun: John cat: snp fnc: know prn: 625		

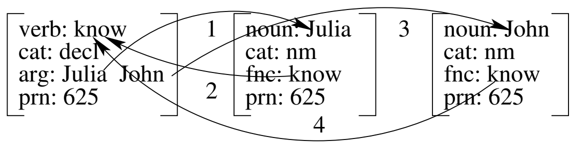
[12] For better drawing, the middle proplet is shown one step to the right.

[11] De Saussure (1916/1972, p. 103) remarked on the time-linear structure of natural language "This principle is obvious, but it seems that stating it explicitly has always been neglected, doubtlessly because it is considered too elementary." This holds still today: most of linguistics is based on the hierarchical principle of possible substitutions, coding dominance and precedence, instead of establishing classic functor-argument and coordination relations in a time-linear derivation order.

The analysis is surface compositional (SCG; FoCL Sect. 4.5; NLC 1.6.1) in that (i) each surface is analyzed as a lexical proplet and (ii) there are no proplets without a surface (lexical lookup). The analysis is time-linear, because each derivation step adds exactly one lexical proplet, as shown by the stair-like appearance. The operations, e.g. **1 NOM×FV** defined in 3.4.2, establish grammatical relations by copying values, as indicated by the diagonal arrows, or by absorbing one proplet into another. As an order free set, the proplets are concatenated by address and constitute a content which is suitable for storage in the agent's content-addressable memory (4.1.1).

Based on the grammatical relations between the proplets stored in the agent's memory, the second step in the cycle of natural language communication is a *selective activation* of content by navigating from one proplet to the next. The following example is based on the content 3.2.1, derived in 3.3.1:

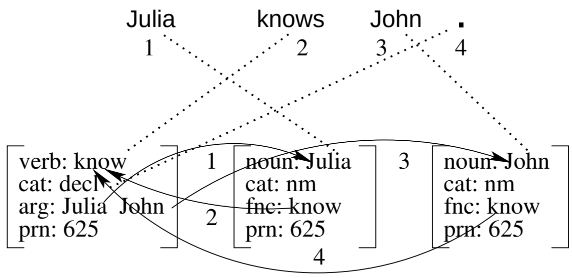
3.3.2 DBS THINK MODE NAVIGATION



The navigation is driven by a DBS.Nav grammar which uses the grammatical relations between proplets like a railroad system. By constructing proplet addresses from the arg, fnc, and prn values, the navigation proceeds from the verb to the subject noun (arrow 1), back to the verb (arrow 2), to the object noun (arrow 3), and back to the verb (arrow 4).

Such a think mode navigation provides the *what to say* for language production from stored content, while the third step in the cycle of communication, i.e. the speak mode, provides the *how to say it* in the natural language of choice (Sect. 1.6). Consider the following example of a speak mode navigation, resulting in a surface realization:

3.3.3 DBS SPEAK MODE REALIZATION



The realization is based on the navigation 3.3.2, whereby the surfaces are realized from the *goal proplet* of each navigation step, using mainly the core value. In NLC, the DBS cycle of communication has been worked out in varying detail for more than 100 constructions of English (cf. NLC Sect. A.10).

3.4 Algorithm of LA Grammar

In earlier publications, the rules of an LA grammar consist of (i) an operation and (ii) a rule package. The function of the rule packages is to control the time-linear order of the rule applications, especially in the complexity analysis of LA grammars for formal languages (TCS; FoCL Part II).

In the LA grammars for natural languages, however, there evolved additional structures in the agent's database, in particular the managing of the now front (Sect. 13.3). This reduced the function of the rule packages to a point at which they could be replaced by *self-organization* (Kohonen 1988). The resulting LA grammars without rule packages are called DBS grammars. A DBS.Hear operation is triggered by a next word proplet matching its second input pattern and applies if it finds a proplet at the now front matching its first input pattern.

Consider the application of NOM×FV (NLC 13.5.1) to the input proplets *Julia* and *know* in line 1 of 3.3.1:

3.4.1 Applying a DBS.Hear operation

NOM×FV

pattern level

$$\begin{bmatrix}\text{noun: }\alpha\\ \text{cat: N}\\ \text{fnc: }\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{cat: N}'\text{ X VT}\\ \text{arg: }\\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: N}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{cat: \#N}'\text{ X VT}\\ \text{arg: }\alpha\\ \text{prn: K}\end{bmatrix} <$$

$N \epsilon$ {snp, pnp, s3, p3}; $N' \epsilon$ {n′, ns3′, n-s3′}.
If $N \epsilon$ {s3, snp} then $N' \epsilon$ {ns3′, n′}; otherwise, $N' \epsilon$ {n-s3′, n′}. VT ϵ {v, vi}.

$$\Uparrow \qquad\qquad \Downarrow$$

content level

$$\begin{bmatrix}\text{sur: }\\ \text{noun: Julia}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: }\\ \text{mdr: heavy\&}\\ \text{nc: }\\ \text{pc: }\\ \text{prn: 4}\end{bmatrix}_1 \begin{bmatrix}\text{sur: knows}\\ \text{verb: know}\\ \text{cat: n}'\text{ a}'\text{ v}\\ \text{sem: past}\\ \text{arg: }\\ \text{mdr: }\\ \text{nc: }\\ \text{pc: }\\ \text{prn: }\end{bmatrix}_5 \begin{bmatrix}\text{sur: }\\ \text{noun: Julia}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: know}\\ \text{mdr: heavy\&}\\ \text{nc: }\\ \text{pc: }\\ \text{prn: 4}\end{bmatrix}_1 \begin{bmatrix}\text{sur: }\\ \text{verb: know}\\ \text{cat: \#n}'\text{ a}'\text{ v}\\ \text{sem: past}\\ \text{arg: Julia}\\ \text{mdr: }\\ \text{nc: }\\ \text{pc: }\\ \text{prn: 4}\end{bmatrix}_5 <$$

Like the operation in an LA grammar rule, a DBS.Hear operation consists of (i) a name, (ii) a pattern for an *ss* (sentence start), (iii) a pattern for an *nw* (next word), and (iv) a pattern for the resulting next sentence start (*ss*′). A pattern at the operation level matches a proplet at the content level if (a) the attributes of the pattern are a sublist of the attributes of the proplet and (b) the values of the pattern are compatible with the values of the proplet (NLC 3.2.3).

By vertically binding the constant value of the feature [noun: Julia] (content level) to the corresponding variable of the feature [noun: α] (pattern level), the feature [arg: α] of the output pattern provides the value Julia to the continuation attribute arg of the output (language level), and correspondingly for know and β (see <). If a current operation application is successful, automatic word form recognition adds a new proplet[12] to the now front. This may result in a new now front pair matching another operation. Compared to the simplified graphical derivation 3.3.1, NOM×FV (i) has additional cat features which show the canceling of the initial valency position by #-marking and (ii) a condition at the matching frontier which specifies the precise agreement conditions of English by means of variable restrictions.

Next let us consider the operations of DBS.Think. They are of two kinds, (i) selective activation by navigating along the semantic relations connecting proplets by address and (ii) inferences which derive new content from given content by matching activated content with their antecedent (induction) or their consequent (abduction). The following DBS.Nav operation V$\swarrow$N proceeds from a verb to a subject noun. It is shown as it executes navigation step 1 in 3.3.2:

3.4.2 Applying a DBS.Nav operation

$$
\begin{array}{lcccl}
 & \text{i. } \textit{operation name} & & & \\
 & \mathbf{V}\swarrow\mathbf{N} & & & \\
 & \text{ii. } \textit{current proplet} & & \text{iii. } \textit{goal proplet} & \\
\textit{pattern level} & \begin{bmatrix}\text{verb: } \alpha \\ \text{arg: } \beta\ \text{Y} \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{noun: } \beta \\ \text{fnc: } \alpha \\ \text{prn: K}\end{bmatrix} & < \quad \textit{instruction:}\ \text{\#-mark } \beta \text{ in the arg slot of proplet } \alpha \\
\textit{matching and binding} & \Uparrow & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{sur:} \\ \text{verb: know} \\ \text{cat: \#n}'\ \text{\#a}'\ \text{decl} \\ \text{sem: pres} \\ \text{arg: Julia John} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 1}\end{bmatrix} & & \begin{bmatrix}\text{sur:} \\ \text{noun: Julia} \\ \text{cat: snp} \\ \text{sem: nm f sg} \\ \text{fnc: know} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 1}\end{bmatrix} & <
\end{array}
$$

By using the same variables, α, β, and K, in the patterns for the current and the goal proplet, though in different slots, and by binding them to the values know, Julia, and 1 of the input proplet *know*, the pattern for the next proplet provides the information required for visiting the successor proplet, here *Julia*.

[12] Assuming that surface input continues.

The *instruction* shown at the pattern level is short for #-mark *the value bound to* β in the arg slot of *the input matching the* proplet α. The #-marking of the value Julia prevents reapplication of V↙N, as in a loop.

Finally consider DBS.Speak. Defined as a variant of standard DBS.Nav or DBS.Inf grammars, its operations contain a language-dependent *lexicalization rule* (defined in NLC 12.4.3, 12.5.2, and 12.6.1) in the sur slot of the output proplet. Taking a language-dependent counterpart of the proplet's core value as input, the lexicalization rule produces zero, one, or more appropriate word form surfaces.

The following example shows the DBS.Speak variant of N↗V performing navigation step 2 in 3.3.3 as it produces knows with the lexicalization rule lexverb operating in the sur slot of the output proplet pattern (see <):

3.4.3 Applying a DBS.Speak operation

$$
\begin{array}{llcll}
 & \textbf{N}\nearrow\textbf{V} & & & \\
\textit{pattern level} & \begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\alpha\\ \text{mdr: Z}\\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{sur: lexverb}(\hat{\alpha})\\ \text{verb: }\alpha\\ \text{arg: \#X }\beta\text{ Y}\\ \text{prn: K}\end{bmatrix} < & \text{\#-mark }\alpha\text{ in the fnc slot of proplet }\beta\\
 & \text{where Z is \#-marked or NIL.[13]} & & & \\
\textit{matching and binding} & \Uparrow & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{sur:}\\ \text{noun: Julia}\\ \text{cat: snp}\\ \text{sem: nm sg}\\ \text{fnc: know}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix} & & \begin{bmatrix}\text{sur: knows}\\ \text{verb: know}\\ \text{cat: \#ns3' \#a' decl}\\ \text{sem: pres}\\ \text{arg: \#Julia John}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix} < &
\end{array}
$$

The language-dependent counterpart of the core value is provided by the lexicon of the language at hand and represented at the pattern level as the variable $\hat{\alpha}$ (pronounced "hat-alpha"). The result of applying lexverb to $\hat{\alpha}$ is shown in the corresponding slot of the corresponding proplet at the content level. This agent-internal, modality-free surface is used as a blueprint (template, 2.3.1) for one of the agent's language synthesis components. As in the DBS.Nav grammar, the output proplet serves as input for the next operation application.

In its most basic form, inter-agent communication in natural language is based on the Second Mechanism of Communication:

---

[13] This condition prevents the application of N↗V and thus a premature navigation to the verb after beginning a noun phrase with an adnominal modifier, e.g. the *old* dog. For complete DBS.Speak grammars producing small fragments of English see NLC Sects. 12.1 and 14.5; HBTR Sect. A.10; TExer Sect. 6.3.

3.4.4 SECOND MECHANISM OF COMMUNICATION (MOC-2)

> The external time-linear surface order is used for coding semantic relations proplet-internally as addresses (hear mode), while the semantic relations coded inside order-free proplets are used for creating a grammatically meaningful time-linear surface order (speak mode).

Extensive linguistic research, beginning with NEWCAT (1986), has shown that the decoding of grammatical relations from nothing but ordered modality-dependent unanalyzed external surfaces works very well. When the proplets resulting from automatic word form recognition are stored in the word bank, any grammatically meaningful order is lost proplet-externally. A DBS.Nav navigation along the proplet-internally coded interproplet relations (re-)introduces a time-linear order, which is used as the macro word order by DBS.Speak.

3.5 Relating Kinds of Proplets to Traditional Parts of Speech

As a computational data structure, proplets are completely general: a proplet may be defined for any set of attributes and any set of values, and still be processed by suitably adapted DBS operations based on matching between the pattern level and the content level. However, long-term upscaling of a talking robot will only be successful if the basic functionality of natural language communication is modeled linguistically correct (CV, 1.3.1). Let us therefore examine the empirical nature of proplets from a linguistic point of view.

The properties of different kinds of DBS proplets are loosely related to the *parts of speech* (PoS) of traditional grammar. The literature on PoS shows considerable variation, but the parts of speech may be summarized as follows:

3.5.1 TRADITIONAL PARTS OF SPEECH

1. *verb*
   Includes finite forms like sang and non-finite forms like singing or sung of main verbs, as well as auxiliaries like was or had, and modals like could and should. Some traditional grammars treat non-finite verb forms as a separate class called *participle*.

2. *noun*
   Includes common nouns like table and names like Julia. Also, count nouns like book and mass nouns like wine may be distinguished.

3. *adjective*
   Includes determiners like a(n), the, some, all, and my as well as adnominals like little, black, and beautiful. Some traditional grammars treat determiners as a separate class.

4. *adverb*
   Includes adverbial modifiers like beautifully and intensifiers like very.

5. *pronoun*
   Includes nouns with an indexical meaning component such as I, me, mine, you, yours, he, him, his, she, her, hers, etc.

6. *preposition*
   Function words which combine with a noun into a modifier, such as on in [the book] on [the table].

7. *conjunction*
   Includes coordinating conjunctions (parataxis) like and and subordinating conjunctions (hypotaxis) like that (introducing a subject or object sentence) or when (introducing an adverbial sentence).

8. *interjection*
   Includes exclamations like ouch!, greetings like hi!, and answers like yes.

Most traditional grammars postulate eight parts of speech because this is the number adopted by classical Greek and Latin grammars.

The parts of speech provide only rough classifications. Also, the term *speech* does not fit nonlanguage (context) content. Yet a computational treatment of reference by means of pattern matching is facilitated if language and nonlanguage content are formalized alike (Sect. 1.6; FoCL Sect. 22.2).

For a characterization of content which is (i) precise and (ii) suitable for both levels, DBS reanalyzes the parts of speech by assimilating them into kinds of proplets. Thereby the terms noun, verb, and adj from grammar are used as the core attributes of functor-arguments, and the terms arg(ument), fnc(functor), mdr (modifier), and mdd (modified) from symbolic logic are used as the continuation attributes.

To accommodate their use at the context level, the core and continuation attributes have an ontological interpretation in DBS. It is similar to the terms *object, relation,* and *property* of philosophy (FoCL 3.4.1), in contradistinction to the parts of speech, which have a grammatical interpretation.

The differentiated attribute structure of only three basic kinds of proplets proved sufficient for exploratory analyses of the following languages:

3.5.2 List of languages analyzed in DBS

Albanian	Kabashi (2003, 2007)
Arabic	ben-Zineb (2010)
Bulgarian	Ivanova (2009), Sabeva (2010)
Chinese	Oberhofer (1993), Mei (2007), Graf (2011), Seneschall (2011), Zhen (2011), Zischler (2011)
Czech	Hučinova
English	Leidner (2000), Bauer (2011), Greiner (2011)
French	Pepiuk (2010), Kosche (2011))
German	Helfenbein (2005), März (2005), Stefanskaia (2005), Twiggs (2005), Girstl (2006), Gao (2007), Mehlhaff (2007), Schörner (2008), Tittel (2008), Handl (2010, 2012), Jaegers (2010), Reihl (2010), Weber et al. (2010), Schneider (2011)
Georgian	Fornwald (2012)
Italian	Wetzel (1996), Weber (2007, 2011)
Japanese	Pankovets (2011)
Korean	Lee (2004), Kim (2009)
Polish	Niedobijczuk (2010), Odrzywolski (2012)
Portugese	Rudat (2011)
Romanian	Pandea (2010)
Russian	Vorontsova (2007), Kalender (2010)
Spanish	Mahlow (2000), Huezo (2003)
Swedish	Lipp (2010)
Tagalog	Söllch (2009)

The analysis of these languages resulted in running software using programming environments written in C (Malaga, Beutel 2009) and Java (Kycia 2004; JSLIM, Handl 2008, 2012). In most cases, especially for the more distant languages, the analysis was done by native speakers. If the analysis of a language is limited to automatic word form recognition it is nevertheless based on detailed studies of the syntax and semantics.

The reanalysis of the traditional parts of speech as kinds of proplets resulted in various changes. For example, traditional grammar classifies 2. *noun* and 5. *pronoun* as different parts of speech. In DBS they have the same core attribute noun because a noun like Julia and a pronoun like she are both (i) elementary, and (ii) may serve the same grammatical functions, e.g. as subject or object.

The correct distinction between common nouns, pronouns, and names is of a sign-theoretic nature: they use different mechanisms of reference (NLC 2.6.4, 2.6.5, 2.6.7; HBTR Sects. 3.2–3.4), namely those of matching, pointing, and baptism establishing coreference by address. Given that common nouns and names, despite their different reference mechanisms, are traditionally included in the part of speech *noun*, there is no reason to exclude pronouns just because they use yet another mechanism of reference, namely pointing.

Consequently, in DBS all three kinds of nouns are analyzed as proplets with the same core attribute. This is illustrated by the following proplets for a common noun, a pronoun, a name, and a determiner:

3.5.3 Analyzing different kinds of nouns as lexical proplets

common noun	*pronoun*	*name*	*determiner*
sur: books	sur: they	sur: Julia	sur: every
noun: book	noun: pro3	noun:	noun: n_1
cat: pn	cat: p3	cat: snp	cat: sn′ snp
sem: pl	sem: pl	sem: nm sg f	sem: pl exh
fnc:	fnc:	fnc:	fnc:
mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:
prn:	prn:	prn:	prn:

The common noun proplet has a core value defined as a concept which is represented as book, serving as a placeholder (6.6.8); following Peirce, word forms with a concept as their content (meaning$_1$) are called *symbols*. The pronoun proplet has a core value defined as a pointer which is represented as pro3;[14] word forms with a pointer as their content are called *indexicals*. The lexical *name* proplet has a sur value, but no core value; the core value is provided in an act of baptism and consists of the address of the referent (HBTR Sect. 3.4).

Some traditional grammars treat determiners as a separate part of speech. The DBS reanalysis of determiners as noun proplets is designed to facilitate the fusion of a determiner and its noun (7.2.4, 7.2.5). As illustrated in 3.5.3, the lexical core value of a determiner is a substitution variable, here n_1; during interpretation in the hear mode, this variable is replaced with the core value of the associated common noun. In some languages, e.g. German, definite determiners are also used indexically, which may be handled by widening the restriction on the substitution variable to include indexical use.

Another difference concerns the treatment of modifiers. The traditional part of speech classification 3.5.1 differentiates between 3. *adjective*, 4. *adverb*, and 6. *preposition*. DBS, in contrast, treats them as proplets which share the core attribute adj and the continuation attribute mdd (modified) and relegates the differences to cat and sem values. Consider the following examples:

[14] See Chap. 11 for the indexical vs. coreferential interpretation of 3rd person pronouns.

3.5.4 Analyzing different modifiers as lexical proplets

adnominal	*adverbial*	*indexical adjective*	*preposition*	
sur: beautiful	sur: beautifully	sur: here	sur: on	
adj: beautiful	adj: beautiful	adj: idx_loc	noun: n_1	
cat: adn	cat: adv	cat: adnv	cat: adnv	
sem: pad	sem: pad	sem:	sem: *on*	
mdd:	mdd:	mdd:	mdd:	<
mdr:	mdr:	mdr:	mdr:	
nc:	nc:	nc:	nc:	
pc:	pc:	pc:	pc:	
prn:	prn:	prn:	prn:	

The proplets beautiful and beautifully have the same core value beautiful (concept), but differ in their cat values adn (adnominal) and adv (adverbial). The proplets here and on share the cat value adnv (adnominal and adverbial use),[15] but differ in the sign kind of their core values, namely pointer (idx_loc) in *here* and the substitution variable n_1 in *on*. Like determiners, prepositions have the core attribute noun (7.2.5), which facilitates the treatment of such different phrases as in Paris, in the city, in the big old city, in the big old city on the river, etc. The particular preposition, e.g. in, on, above, below, etc., is specified as the first value of the sem attribute.

Analyzing the adnominal and adverbial uses of adjective proplets with the same core attribute, i.e. adj, is partially motivated by terminology: the Latin root of *adjective* means "what is thrown in," which aptly characterizes the optional nature of modifiers in general. It is also motivated morphologically because of the similarity between adnominal (beautiful) and adverbial adjectives (beautifully). The two uses may resemble each other also in their analytic degrees, as in more beautiful (adnominal) and more beautifully (adverbial) in English.

Languages may differ in whether they treat the adnominal or the adverbial form of an adjective morphologically as the unmarked case. For example, in English the unmarked case is the adnominal,[16] e.g. beautiful, while the adverbial is marked, e.g. beautifully. In German, in contrast, it is the reverse: the adnominal is marked, e.g. schöner, schöne, schönes, etc., while the adverbial, e.g. schön, is unmarked.

The relations between English modifiers with the cat values adnv, adn, and adv may be shown graphically as follows (NLC Sects. 15.1–15.4):

[15] As a cat value, adnv is an empirically motivated underspecification in DBS. For a comparison of underspecification in DBS and in nativism see NLC p. 91, footnote 5.

3.5.5 Relation between the adnv, adn, and adv values in English

modifier [cat: adnv]
on the table,
here, fast

adnominal [cat: adn]
beautiful

adverbial [cat: adv]
beautifully

Modifiers which can modify nouns as well as verbs have the cat value adnv. This value characterizes the combinatorics of prepositional phrases like on the table, indexical adjectives like here, and certain symbol adjectives like fast. If an adjective has two forms, e.g. beautiful and beautifully, the unmarked form has the cat value adn (adnominal use) in English, while the marked form has the cat value adv (adverbial use).

Finally, we turn to the traditional parts of speech which share the core attribute verb in DBS. These are finite verbs, auxiliaries, and participles:

3.5.6 Analyzing different verb forms as lexical proplets

finite main verb	*finite auxiliary*	*non-finite main verb*
sur: knows	sur: is	sur: knowing
verb: *know*	verb: v_1	verb: *know*
cat: ns3′ a′ v	cat: ns3′ be′ v	cat: a′ be
sem: ind pres	sem: ind pres	sem: prog
arg:	arg:	arg:
mdr:	mdr:	mdr:
nc:	nc:	nc:
pc:	pc:	pc:
prn:	prn:	prn:

In declarative and echo Yes/No interrogative sentences, finite verb forms are characterized by a cat value ending with the constant v. This holds for main verbs and auxiliaries alike. The difference between the two is that English auxiliaries, but not the main verbs, have be′, hv′, or do′ as one of their cat values (NLC 6.6.8, FoCL Sect. 17.3). Also, the core value of an auxiliary (function word) is a substitution value, here v_1 (NLC Sect. 6.3), while that

[16] That the adnominal form happens to equal the unmarked case of adjectives in English may be the reason for the widespread but nevertheless misguided terminological practice of calling the adnominal use "adjective," in contrast to the adverbial use, which is called "adverb." If we were to apply the same logic to German, it would have to be the adverbials which are called "adjective."

Equating adnominal with adjective reduces the number of available terms from three to two. However, three terms (3.5.5) are required by the combinatorial possibilities concretely reflected in the morphology (surface compositionality). The adj value adnv is used for expressions without any morphological restrictions for an adnominal or an adverbial use (fast), adn for adnominal expressions which are morphologically restricted to noun modification (beautiful), and adv for adverbial expressions which are morphologically restricted to verb modification (beautifully).

of a main verb is a concept, here know. Non-finite verb forms differ from finite ones by the absence of the constant v at the end of their cat value. The verbal moods of the *indicative, subjunctive,* and *imperative* are coded as the sem values ind, sbjv, and impv, respectively (Sect. 5.6; NLC A.3.2, 8).[17]

There remain the traditional parts of speech *7. conjunction* and *8. interjection*. Conjunctions are function words which are used for connecting expressions of the same part of speech (coordination, parataxis, NLC Chaps. 8 and 9) or of different parts of speech (subordination, hypotaxis, NLC Chap. 7). Interjections are one-word sentences; their lexical analysis is not constrained by the compositional considerations of establishing semantic relations and they still await analysis as proplets in Database Semantics.

In summary, despite the use of only three core attributes in DBS, the traditional parts of speech may be reconstructed using other properties. For example, based on the lexical analysis of word forms in 3.5.3, *pronouns* as a traditional part of speech are characterized by a pointer core value, while *determiners* have a substitution variable instead. Similarly, based on the lexical analysis in 3.5.4, *prepositions* as a traditional part of speech are characterized by the cat value adnv and a substitution variable as their core value, adnominals by the cat value adn and a concept as their core value, indexical adverbs by the cat value adv and a pointer as their core value, and so on.

Such grammatical distinctions may be easily used to lexically define word forms as proplets which have precisely the properties desired: if available, we choose the appropriate terms from traditional grammar for the attributes and values. Otherwise we coin new terms which are as informative as possible. These grammatical distinctions are used in automatic pattern derivation. Each pattern constitutes a particular *view* on the data (4.5.5), providing for a highly differentiated retrieval from the agent's memory (database, word bank).

3.6 Linguistic Relativism vs. Universal Grammar

Before turning to the differences between natural languages let us summarize what they have in common. The language-independent (universal) properties of a DBS system are based on the external interfaces of the agent, the data structure of proplets, the database schema of a content-addressable memory storing content consisting of sets of proplets concatenated by address, and the time-linear algorithm of DBS grammars for mapping between raw data and

[17] For a more detailed discussion of verbal moods see Quirk et al. (1985:149–150).

content as well as for processing content. This component structure works for language and nonlanguage cognition alike.

The core attributes of proplets are noun, adj, and verb. Core values use one of the three reference mechanisms of natural language, namely *concept* (symbol), *pointer,* (indexical) and *baptism* (name). Thereby the core attribute noun may take a concept (common noun), a pointer (pronoun), or a name as value; the core attribute adj may take a concept or a pointer; and the core attribute verb may take a concept only (NLC 2.6.8, Seventh Principle of Pragmatics). The DBS notion of generalized reference (HBTR 3.1.2) works with content and therefore does not require language.[18]

The continuation attributes are fnc, arg, mdr, mdd, nc, and pc. They characterize the goal proplet of a semantic relation as a functor, an argument, a modifier, a modified, a next conjunct, and a previous conjunct, respectively. The semantic relations of structure, i.e. classic functor-argument and coordination, are used for building language and nonlanguage content alike.

A continuation value is defined as the address of another proplet, i.e. the goal proplet of a semantic relation of structure. An address consists of (the name of) the core value and the prn number. In an intrapropositional relation, the core value alone is sufficient (*short address*) because the prn value of the address equals the prn value of the proplet, e.g. Julia in 4.4.1. In an extrapropositional relation, a *long address* is used, e.g. (Julia 675) in 4.4.3.

The grammatical role of a proplet results from the core and the continuation attributes and values of the proplets connected within a content. There are six basic roles in natural language, namely subject, predicate, object, adnominal, adverbial, and conjunct. For example, in 7.3.2 the subject role of The little girl is realized by writing girl into the initial arg slot of *eat*, while the object role of an apple is realized by writing apple into a noninitial arg slot of *eat*.

The grammatical roles arise at the elementary, phrasal, and clausal level of grammatical complexity. Depending on the level, a grammatical role may be represented by proplets with different core attributes. For example, the core attribute of an adnominal modifier is adj at the elementary level (NLC 6.2.1), noun at the phrasal level (NLC 15.2.1), and verb at the clausal level (NLC 7.3.2).

The constructs of (i) ontological role (core attribute), (ii) reference mechanism (core value), (iii) semantic relation of structure (continuation attribute), (iv) address (continuation value), (v) grammatical role, and (vi) level of gram-

[18] Reimer and Michaelson (2014) use the term “representational token” for defining reference. This seems to allow for reference without language, like our notion of content.

matical complexity are used for building and processing *content*. So where are the differences between natural languages (16.4.1)?

The most obvious difference are the sur values. This difference is mitigated in those cases in which different languages use the same core values in content words. For example, the lexical proplets for English dog, French chien, German Hund, Italian cane, and Polish pies have the same core value, here the concept *dog* (6.6.3). Similarly, grammatical values for different numbers, tenses, genders, honorifics, etc., may be used across those languages which use these distinctions. The language-dependent word order (16.4.2)[19] does not affect the content coded by different languages because it is dissolved by the hear mode into an order-free set of proplets (3.2.7)).

The one place in which natural languages differ apart from their sur and certain cat and sem values are different lexicalizations,[20] e.g. different valency structures, and different syntactic-semantic constructions[21] for coding the same content. This applies not only to languages which are distant from each other, such as English and Chinese, but also to neighboring languages such as English, German, and Italian.

For example, the following sentences have equivalent meanings$_1$ (1.4.1), but their syntactic-semantic structure in English, German, and Italian is different:

3.6.1 Equivalent clauses with different constructions

English:	I don't care.	
German:	Es ist mir egal.	(*It is me equal*)
Italian:	Mi lascia indifferente.	(*Me leaves-it indifferent*)

In the English variant, the subject position may be filled by any non-clausal noun such as phrasal the man with the brown coat or elementary you and John, with the finite verb form suitably adapted. The finite verb is the auxiliary do in its negated form and may take different tense forms. The main predicate is the infinitive of the verb care, which could be replaced by the verb mind without a substantial change of the meaning$_1$.

---

[19] For a uniform method of mapping the different word orders in declarative main clauses of German (second position of the finite verb with free order of the obligatory arguments), English (postnominative position of the verb with fixed order of the obligatory arguments), Korean (final position of the verb with free order of the obligatory arguments), and Russian (free order of the verb and its arguments) into equivalent contents, see Hausser (2008). The order of optional modifiers is usually comparatively free.

[20] The literature on lexicalization is vast. See Talmy (1985); Jackendoff (1990); Pustejovsky (1995); Brinton and Traugott (2005), and others. Lexicalization is perhaps the only branch of contemporary linguistics in which the crucial distinction between the literal meaning$_1$ of a sign and the speaker meaning$_2$ of an utterance is more or less properly observed (PoP-1, FoCL 4.3.3).

[21] This notion is central to construction grammar (Fillmore et al. 1988; Croft 2000). The syntactic-

In the German variant, the subject is expletive es (pronoun).[22] The finite verb is the auxiliary sein (corresponding to English be) without negation, which may take various tense and mood forms. The counterpart to the subject of the English clause is any noun in the dative case. The main predicate is the adverbial egal, which could be replaced by gleich or wurst without changing either the form or the meaning.[23] And correspondingly for Italian.

Such asymmetries between different natural languages have long been noted and have fostered a tradition within language theory. It is known as linguistic relativism or the Humboldt-Sapir-Whorf hypothesis, according to which there does not exist a universal (natural, innate) semantics which all natural languages map into and out of[24] (interlingua, 16.4.7).

In the extreme, even the part of speech distinctions corresponding to the DBS core attributes noun, verb, and adj have been claimed to be absent in some natural languages. Specifically, the Iroquois language *Cayuga* has been argued to have no distinction between verbs and nouns (Sasse 1993), though Hengeveld (1992), Rijkhoff (2002), and Wunderlich (2006) disagree.[25]

For the outsider, such controversy is difficult to assess.[26] First, for one not speaking Cayuga natively, it is most precarious to judge the empirical facts. They concern not only the surfaces provided by the tape recorder, but also the intuitions of the speaker-hearer.

Second, there is the theoretical question of whether the claim applies to morphology, syntax-semantics, or pragmatics. As a related phenomenon consider conversion in English: a surface like makes can be used as a noun in Jim collects all makes and as a verb in Jim makes toys.[27]

By analogy, the claimed non-distinction between nouns and verbs in Cayuga may apply to (i) nouns and verbs sharing a stem, like make in English, or (ii) nouns and verbs sharing a stem and morphological alternations, like

---

semantic analysis of constructions is no different from that of any other grammatical language expression. It is just that constructions seem to be of a more collocational and/or idiomatic nature than spontaneous phrasing (NLC Sect. 15.6).

22 Grammatically acceptable subject alternatives for expletive es, such as der Preis (the price), would compromise the equivalence with the English counterpart.

23 Intensified versions of egal in German are schnurzegal and schnurzpiepegal. The piep in the latter has been suggested to be a stand-in for a swearword– perhaps as an early form of the "censor beep"?

24 For a modern argument from language typology see Nichols (1992).

25 Mithun's (1999) formidable overview of the Iroquois languages lists the paper by Sasse in the bibliography, but does not discuss or cite it in the text.

26 At last count, Cayuga is spoken by fewer than one hundred native Americans living in Ontario, Canada. Such a small base of native speakers makes it especially difficult to obtain reliable scientific information about the language.

27 The DBS system for English treats the concept *make* as a core value which may be embedded into the proplet shell of a noun or a verb (see 6.6.5 – 6.6.7 for related examples).

make/make+s in English, or it may mean (iii) that the native speakers of Cayuga do not distinguish between objects and relations. Position (iii) corresponds to the radical linguistic relativism of Whorf (1964).

The position of DBS is a moderate linguistic relativism. The moderation consists in treating all natural languages as functioning the same (Sect. 1.2; NLC 4.6.1). The only essential differences between natural languages which the DBS software machine has to deal with arise in lexicalization, collocation, and idioms – with subsequent ramifications in syntactic-semantic composition.

Thus translating a content from a construction in one language to a different construction in another is essentially a matter of effort. Coding differences may be overcome by patching them with paraphrases and by adding background information (Sect. 16.5). However, if qualities of the surface and/or culture-dependent associations are important, as in poetry (Sect. 16.4), reconstruction in another language can at best be an approximation.

4. Mystery Number Three: Memory Structure

Because the relations between proplets within a proposition and between propositions are coded solely by means of addresses defined as proplet-internal continuation values, proplets are inherently *order-free* (3.2.8). Thus proplets may be stored in accordance with any kind of database schema, without losing their interproplet relations, but gaining the use of the storage and retrieval mechanism of the database selected. Given that computer science offers different database schemata, the question of the third mystery is: which schema is suited best for modeling cognition, including the cycle of natural language communication.

4.1 Database Schema of a Word Bank

When faced with the choice of a computer memory, the most basic alternative is between a (i) *content*-addressable and a (ii) *coordinate*-addressable approach (see Chisvin and Duckworth 1992 for an overview). A coordinate-addressable memory, e.g. an RDBMS, resembles a modern public library in which a book may be stored wherever there is space (random access) and retrieved using a separate index (inverted file) relating a primary key (e.g. author, title, year) to its location of storage (e.g. 1365).

A content-addressable memory, in contrast, is like a private library in which books with certain properties are grouped together on certain shelves, ready to be browsed without the help of a separate index. For example, at Oxford University the 2 500 volumes of Sir Thomas Bodley's library from the year 1598 are still organized according to the century and the country of their origin. Computationally, a content-addressable memory is suited best for the super-fast retrieval of content which is written once and never changed.

Though peppered with patents, the content-addressable approach is currently less widely used than the coordinate-addressable approach. In an initial response to the content-addressable approach of DBS, mainstream database scientists pointed out that it can be simulated by the coordinate-addressable approach (Fischer 2002), using well-established relational databases. The issue here, however, is whether or not the formal intuitions of the content-addressable approach may be refined naturally into a model of cognition.

Our point of departure for designing the database schema of DBS is the data structure of proplets (Sect. 3.1). For storage and retrieval, a proplet is specified uniquely[1] by its core and prn values (primary key). This suggests a two-dimensional database schema, as in a classic network database (Elmasri and Navathe 2010). However, instead of using member and owner records, DBS uses member proplets and owner values.

The result is called a word bank. Its database schema consists of a column of owner values in alphabetical order (vertical). Each owner value is preceded by an empty slot, called the now front, and a list of member proplets (horizontal); together they constitute a *token line*.

As an example, consider storing the proplets of the content 3.2.1:

4.1.1 Storing the proplets of 3.2.1 in a word bank

	member proplets		*now front*	*owner values*
...				...
...	$\begin{bmatrix}\text{noun: john}\\ \text{cat: snp nm m}\\ \text{fnc: buy}\\ \text{prn: 610}\end{bmatrix}$	$\begin{bmatrix}\text{noun: john}\\ \text{cat: snp nm m}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix}$		john
...				
...	$\begin{bmatrix}\text{noun: julia}\\ \text{cat: snp nm f}\\ \text{fnc: read}\\ \text{prn: 605}\end{bmatrix}$	$\begin{bmatrix}\text{noun: julia}\\ \text{cat: snp nm f}\\ \text{fnc: know}\\ \text{prn: 625}\end{bmatrix}$		Julia
...				...
...	$\begin{bmatrix}\text{verb: know}\\ \text{cat: ... interrog}\\ \text{arg: peter suzy}\\ \text{prn: 608}\end{bmatrix}$		$\begin{bmatrix}\text{verb: know}\\ \text{cat: \#s3}'\ \text{\#a}'\ \text{decl}\\ \text{arg: julia john}\\ \text{prn: 625}\end{bmatrix}$	know
...				...

The proplets in a token line all have the same core value and are in the temporal order of their arrival,[2] as reflected by their prn values (Sect. 13.3).

In contrast to the task of designing a practical schema for arranging the books in a private library,[3] the sorting of proplets into a word bank is simple and mechanical. Any new arrival is stored at the penultimate position (now front) in the token line corresponding to the proplet's core value. When this slot is

---

[1] Propositions containing two or more proplets with the same values, as in Suzy loves Suzy, require extra attention. They constitute a special case which (i) rarely occurs and (ii) is disregarded here because it is easily programmed.

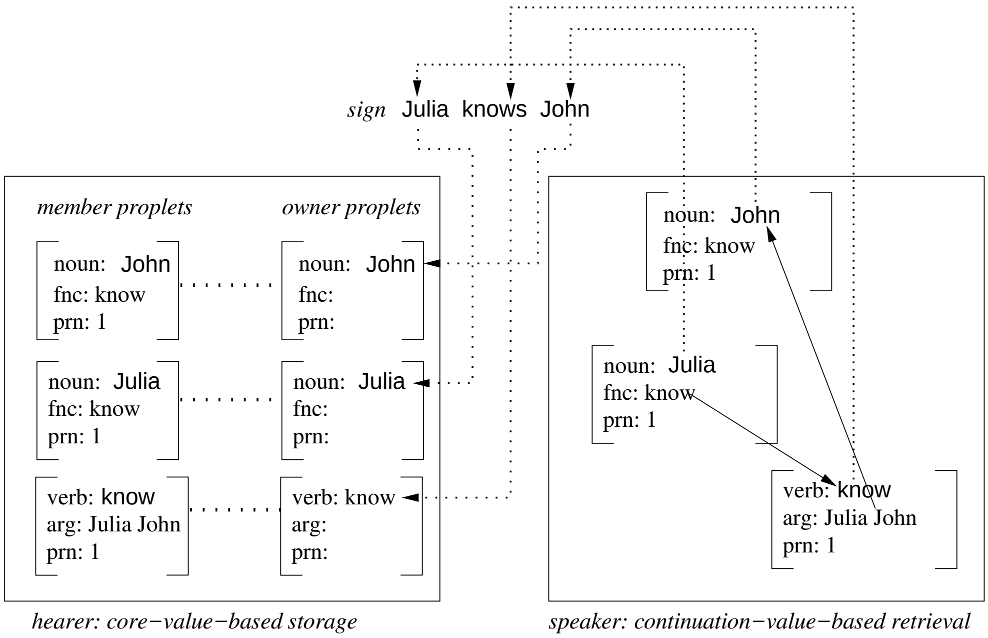
[2] The token line for any owner value is found by using a trie structure (Briandais 1959, Fredkin 1960). The search for a proplet within a token line may use the prn value of the address in relation to the strictly linear increasing prn values. As pointed out by J. Handl, this may be based on binary search, in time $O(\log(n))$ (Cormen et al. 2009), or interpolation, in time $O(\log(\log(n)))$ (Weiss 2005), where n is the length of the token line.

filled, the now front is reopened by moving the owner value one slot to the right (or, equivalently, pushing the member proplets one slot to the left).

This way of storing content like *sediment* completely precludes modification of the stored data (Sect. 14.5) and thus obviates the usual consistency checking (Schewe and Thalheim 1994). If there are changes of fact, they are written to the now front, like diary entries recording changes of temperature. New data are related to old ones by means of addresses, implemented as pointers.

In natural language communication, the database schema of a word bank (4.1.1), the data structure of proplets, and the DBS algorithm for the speak and the hear mode interact as follows:

4.1.2 DBS MECHANISM OF CONTENT TRANSFER WITH LANGUAGE



The speaker's word bank contains the content 3.2.1, sorted into token lines. The DBS.Speak algorithm (e.g. 3.4.3) navigates along the semantic relations connecting the proplets (simplified compared to 3.3.3). The core values of the proplets traversed are passed to the agent's action component and realized as ordered modality-dependent unanalyzed external surface tokens (Chap. 2).

The hearer's recognition component uses the incoming external surfaces (*sign*) for lexical lookup (2.5.1). The lexical proplet of a next word is stored in the appropriate token line at the now front and connected by the time-linear

[3] Applied to the storage/retrieval schema of Sir Thomas Bodley's library, the core value would be the country of origin and the prn value the century of a book's production.

DBS.Hear derivation shown in 3.3.1. When the now front in a token line is cleared by moving the owner value one step to the right, most (4.1.1) of the connected proplets are left behind in the field of member proplets (sediment). The information transfer illustrated in 4.1.2 is successful insofar as the hearer reconstructs the speaker's content using nothing but the time-linear sequence of modality-dependent unanalyzed external word form surfaces.

A word bank is content-addressable because no separate index (inverted file) is being used. Furthermore, a word bank is scalable (a property absent or problematic in some other content-addressable systems). The cost of insertion is constant, independent of the size of the stored data, and the cost of retrieving a specified proplet grows only logarithmically with the data size (external access) or is constant (internal access). External access to a proplet requires (i) its core and (ii) its prn value, e.g. (know 625). Most cognitive operations, however, require internal access based on addresses (pointers; Sect. 4.4).

Compared to the 1969 CODASYL network database, a word bank is highly constrained. First, the member proplets belonging to an owner value are listed in the temporal order of their arrival. Second, the members in a token line must share the owner value as their core value (no multiple owners). Third, the only connections between proplets across token lines are the classic semantic relations of functor-argument and coordination. Fourth, like the relations between owners and members, the semantic connections are 1:n relations: one functor – several possible arguments; one first conjunct – several possible successors; one original – several possible coreferent address proplets.

A word bank is a kind of navigational database because it supports the navigation from one proplet to the next, using the semantic relations across (3.4.3) and along token lines (4.2.2) like a railroad system and the operations of DBS like a locomotive. The complete system is located inside the cognitive agent which serves as the container and structural foundation of autonomous control (Chap. 5). This is in contradistinction to the navigational databases of the past (Bachman 1973) and the present (XPath, Kay 2004), which are designed to be driven by system-external human users.

4.2 Retrieving Answers to Questions

In the hear mode, lexical proplets produced by automatic word form recognition are (i) stored at the now front according to their core value and connected *in situ* (Sect. 13.3) by a syntactic-semantic derivation (3.3.1). The inter-proplet connections support (ii) navigating from a given proplet to a successor proplet (4.1.2) across token lines as selective activation (think mode). Because the speak mode is riding piggyback on the think mode (3.3.3, 3.4.3), the proplets in a word bank are suitable (iii) for language production as well.

Another word bank operation is (iv) retrieving answers to questions (Sects. 10.4, 10.5). This operation is based on moving a query along token lines until matching between the query and a content is successful. A query is a pattern, i.e. a set of connected proplets with at least one variable as value.

Consider an agent wondering who is reading what. This means activating the token line of *read*, as in the following example:

4.2.1 Example of a token line

member proplets				*now front*	*owner value*
verb: read arg: Bill sports prn: 10	verb: read arg: John politics prn: 12	verb: read arg: Mary novel prn: 15	verb: read arg: Suzy stocks prn: 19		read

As indicated by the arg and mdr values of the member proplets, the agent happened to observe or hear about Bill reading sports, John reading about politics, Mary reading a novel, and Suzy reading the stock market results.

For retrieval, the member proplets of a token line may be checked systematically by moving a query from right to left. The following example shows a query consisting of a single pattern proplet representing the query Who is reading sports? as it applies to the token line 4.2.1:

4.2.2 Applying a query pattern to a token line

		query pattern	verb: read arg: α sports prn: K		
member proplets			*matching?*	*now front*	*owner value*
verb: read arg: Bill sports prn: 10	verb: read arg: John politics prn: 12	verb: read arg: Mary novel prn: 15	verb: read arg: Suzy stocks prn: 19		read

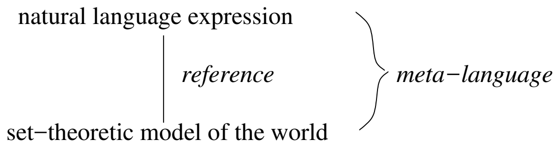
The relevant token line is found via the core value of the pattern proplet.

The attempt at matching indicated by ? fails because the second arg value of the pattern proplet (i.e. sports) and of the member proplet (i.e. stocks) are incompatible. The same holds after moving the pattern proplet one member proplet to the left. The matching attempts continue to be unsuccessful until the leftmost member proplet is reached. Now the variable α is bound to the value Bill and the variable K to the value 10. Accordingly, the answer provided to the question Who reads sport? is Bill (NLC Sect. 5.1). The use of queries consisting of several concatenated pattern proplets (e.g. 3.2.1–3.2.6, 5.2.4, 6.5.1) is based on coactivation and intersection (Sect. 5.4).

4.3 Reference as an Agent-Internal Cognitive Procedure

In the tradition of medieval logic, today's analytic philosophy and linguistics define reference as a direct relation between signs, e.g. language surfaces, and the world (sign-based approach), thus by-passing the cognitive operations of the communicating agents. For example, model theory formally reconstructs reference by defining set-theoretical models which relate[4] language signs directly to an abstraction of "the world":

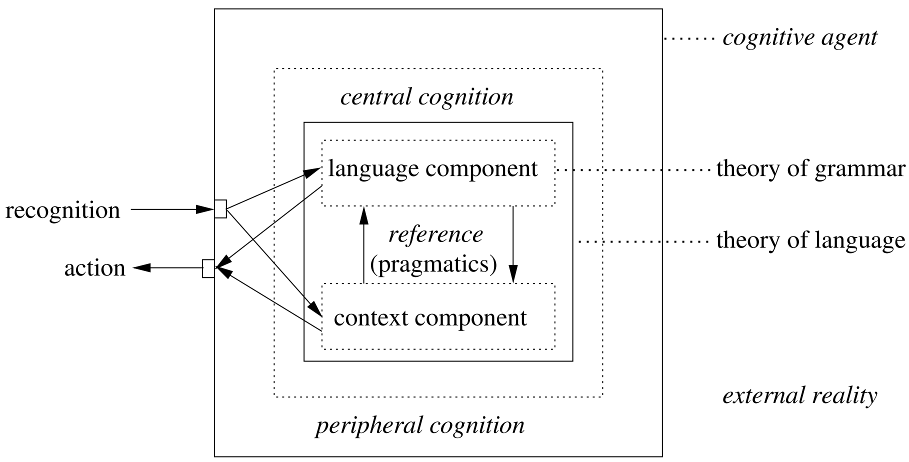
4.3.1 Sign-based reconstruction of reference



The language expression, the world, and the reference relation between the expressions and the world are all defined in a metalanguage – without any cognitive agents and therefore without any agent-internal database. For the design of a talking robot, however, the external environment (i.e. the "world") is a given, to be recognized and acted in by the agent. This requires a procedural approach (Schnelle 1988; FoCL Sect. 19.4) to reference in terms of efficient software operations.

In agent-based DBS, reference is designed as a cognitive process. It is based on a component structure[5] which goes back to Hausser (1980):

4.3.2 Agent-based reconstruction of reference



---

[4] For example, by defining the denotation function F in Montague (1974), PTQ.
[5] NLC Sects. 2.4, 2.5; for further refinements see 4.5.3, 4.5.4, and 14.3.2.

The agent, natural or artificial, is represented as a body in the real world with interfaces for recognition and action. Externally, the interfaces for language and nonlanguage recognition are the same, as are those for language and nonlanguage action. Internally, however, the language and context data are separated by peripheral cognition (diagonal arrows). Thus, in recognition raw input data may result in a language or a nonlanguage content, and in action a content is realized as raw output regardless of whether it originated at the language or the nonlanguage level.

It took 35 years to complete a computationally viable analysis[6] of the agent-internal pattern matching between the language and the context component as shown in 4.3.2. The solution is based on (i) using the same data structure (proplets) at the language and the context level, (ii) using the same matching between pattern and content proplets for the operations of the hear mode(3.4.1), the think mode (3.4.2), querying (4.2.2), and the speak mode (3.4.3), and (iii) using the same semantic relations, namely functor-argument, coordination, and coreference, for language and non-language content. These are integrated into a cognitive agent which uses (iv) the same interfaces, (v) the same time-linear derivation order, (iv) the same storage at the now front, and (vi) the same retrieval by address for language and nonlanguage content.

The initial idea for a computational pattern matching between the language and the context level came from philosophy, namely the use of the type-token relation (8.5.1, 8.5.3). The following example is based on proplet structures:

4.3.3 REFERENCE AS LANGUAGE-CONTEXT PATTERN MATCHING

types

language level	sur: der Hund noun: dog fnc: bark prn: 23	sur: bellte verb: bark arg: dog prn: 23
	reference	
context level	sur: noun: dog fnc: bark prn: c16	sur: verb: bark arg: dog prn: c16

tokens

[6] Neither the formulas of typed lambda calculus in SCG (Montague grammar) nor the trees generated from frames in CoL (FrameKit, Carbonell and Josef 1986) proved to be practical for implementing a computational pattern matching between a language component and a context component (Sect. 12.4; FoCL Sect. 22.2.) Today's solution was first described in a 2015 preprint of the Second NLC Edition.

The proplets at the context level use core value tokens to represent the content *dog bark*. The proplets at the language level use core value types to code the same content and have German surfaces in their sur slot.

The pattern matching is successful because (i) corresponding proplets have the same attributes in the same order and (ii) corresponding values are compatible. The sur values are compatible because the context proplets have the sur value NIL, represented as empty space in the value slot, i.e. [sur:].[7]

The pattern matchings used (i) in the application of operations (3.4.1, 3.4.2, 3.4.3) and (ii) in establishing reference (4.3.3) differ in the kinds of values which have to be compatible. In the pattern-content matching of the operations, the matching values are restricted variables in the patterns and constants in the content (4.5.2). In language-context matching with the sign kind symbol,[8] in contrast, the matching core values are concept types at the language level and concept tokens at the context level (4.5.1, Sect. 8.5; NLC Sect. 4.1).

4.4 Coreference by Address

Computer science uses the term “reference” differently from today’s analytic philosophy and linguistics. A computational reference is the address of a storage location. This may be coded (i) as a *symbolic address* (declarative) or (ii) as a *pointer* to the physical storage location in the memory hardware (procedural, Sect. 1.5). Because pointers access data items directly by storage location, they are more efficient than symbolic addresses. Pointers provide instant access in constant time, and are especially suited for data which are written once and never changed – as in the content-addressable memory of DBS.

Because pointers to a physical storage location do not lend themselves to a declarative specification, our representation uses symbolic addresses instead. For example, during language interpretation, symbolic addresses are established by means of copying. The following example shows the first derivation step of 3.3.1, with the result of cross-copying but before next word lookup:

4.4.1 SYMBOLIC ADDRESSES RESULTING FROM CROSS-COPYING

1 N+FV	[noun: Julia cat: snp fnc: prn: 625]	[verb: know cat: s3’ a’ v arg: prn:]	*result*	[noun: Julia cat: snp fnc: know prn: 625]	[verb: know cat: #s3 a’ v arg: Julia prn: 625]

While the hear mode operations cross-copy core values as addresses into the slots of continuation attributes, their use in the think mode is in the opposite direction, i.e. from continuation values to core values (3.3.2).

For example, when the proplet *Julia* is activated in the course of a navigation, the symbolic address (know 625), constructed from its fnc and prn values allows the navigation to continue to the relevant *know* proplet via external access: go to the owner value know and look for the member proplet with the prn value 625 in the token line (4.1.1). Computationally, however, internal access is more efficient. It is implemented as a pointer from the fnc value of the *Julia* proplet to the physical storage location of the relevant *know* proplet.

The coding of interproplet relations by means of addresses which are (i) specified declaratively and (ii) implemented as pointers combines a declarative specification with a different but functionally equivalent computational realization of high efficiency. The symbolic addresses continue to be a practical necessity because they allow us to recompute the physical storage locations after occasional clean-ups of the agent's primary memory (Sects. 14.5, 14.6).

Another use of addresses, symbolic and by pointer, is for connecting new content to old content by means of coreference. Consider, for example, a cognitive agent observing at moment t_i that *Julia is asleep* and at t_j that *Julia is awake*, referring to the same person. Instead of representing this change by revising the first proposition into the second, the second proposition is added as new content, like sediment, leaving the first proposition unaltered:[9]

4.4.2 Coreferential coordination in a word bank

	member proplets				*now front*	*owner values*
. . .	[noun: Julia fnc: sleep prn: 675]	. . .	[noun: (Julia 675) fnc: wake prn: 702]	. . .		Julia
. . .						
. . .	[verb: sleep arg: Julia prn: 675]	. . .				sleep
. . .						
. . .		. . .	[verb: wake arg: (Julia 675) prn: 702]	. . .		wake

The core attribute of the Julia proplet with the prn value 702 has a long address value[10] in the core feature [noun: (Julia 675)], instead of the short value in the core feature [noun: Julia] of the initial referent with the prn value 675.

---

[7] Currently, don't_care values and must_be_NIL values are represented alike as empty space. Strictly speaking, this representation must be disambiguated by using different kinds of values.

[8] The three sign kinds of natural language are symbols (in the sense of Peirce), indexicals, and names (FoCL 6.1.1). Symbols provide the place for integrating the prototype theory of Rosch (1999).

[9] In other words, changing original data, e.g. from *fuel high* to *fuel low*, is impossible in DBS.

[10] An address coding an extrapropositional relation is written as (the name of) another proplet's core value followed by its prn number and placed in parentheses, e.g. [arg: (Julia 675)] (long address).

This method, called coreference by address, enables a given item to code as many relations to other proplets as needed, without increasing the number of attributes or values within proplets. For example, the proplets in the token line of *Julia* have the fnc value sleep in proposition 675, but wake in proposition 702. The most recent (and thus most up-to-date) content relating to the initial referent is found by searching the relevant token line from right to left, i.e. in the anti-temporal direction.

Coreference by address establishes an identity relation between the *core* values of two proplets,[11] in contradistinction to functor-argument and coordination, which establish an identity relation between the *continuation* value of one proplet and the *core* value of another. Accordingly, coreference by address relates proplets within the same token line (4.4.2, top line), while functor-argument (4.1.1) and coordination (3.2.4) relate proplets across token lines.

Defining core values as addresses does not obstruct the strictly time-linear order of storing proplets in a token line, yet allows us to refer back to a coreferent proplet within that token line. As a result, coreference by address provides for a third kind of DBS.Nav navigation – in addition to moving along the semantic relations across token lines (3.3.2), and moving a query pattern along a token line (4.2.2). Consider the following example:

4.4.3 COREFERENTIAL NAVIGATION

$$\begin{bmatrix}\text{verb: sleep}\\ \text{arg: Julia}\\ \text{prn: 675}\end{bmatrix} \overset{1}{\leftrightarrow} \begin{bmatrix}\text{noun: Julia}\\ \text{fnc: sleep}\\ \text{prn: 675}\end{bmatrix} \overset{2}{\leftrightarrow} \begin{bmatrix}\text{noun: (Julia 675)}\\ \text{fnc: wake}\\ \text{prn: 702}\end{bmatrix} \overset{3}{\leftrightarrow} \begin{bmatrix}\text{verb: wake}\\ \text{arg: (Julia 675)}\\ \text{prn: 702}\end{bmatrix}$$

Connections 1 and 3 are intrapropositional relations between (order-free) *Julia* and *sleep*, and *Julia* and *wake*, respectively. Connection 2 is extrapropositional and based on the coreference between the address proplet of proposition 702 and the original *Julia* proplet of proposition 675. The content of 4.4.3 may be realized in English as Julia was asleep. Now she is awake.

4.5 Component Structure

In DBS, pattern matching based on the type-token distinction[12] is used for the following applications:

In an intrapropositional relation, in contrast, the goal proplet's core value alone is sufficient as the address, e.g. [arg: Julia] (short address), because the proplets of an elementary proposition all have the same prn value.

[11] Coreference in natural language seems to limited to nouns. The coreference by address in DBS, in contrast, does not exclude verbs. For example, in the sequence Fido barked. ... When Fido barked., the second barked could be represented as coreferent with the first.

4.5.1 Pattern matching based on the type-token relation

a. *Recognition*:
matching between concept types and raw input, resulting in content tokens (Sect. 8.5; NLC Sect. 4.3)

b. *Action*:
matching between concept tokens and concept types, resulting in raw ouput (NLC Sect. 4.4)

c. *Reference*:
matching between language and context proplets (4.3.3; NLC 3.2.4)

Pattern matching based on restricted variables, in contrast, is used apply DBS operations to input content:

4.5.2 Pattern matching based on restricted variables

a. *Natural language interpretation*:
matching between DBS.Hear operations and language proplets (3.4.2)

b. *Navigation*:
matching between DBS.Nav operations and content proplets (3.4.3)

c. *Language production from non-language content*:
matching between DBS.Speak operations and content proplets (3.4.3)

d. *Querying*:
matching between query patterns and content proplets (4.2.2)

e. *Inferencing*:
matching between DBS.Inf operations and content proplets (5.2.4, 5.3.4)

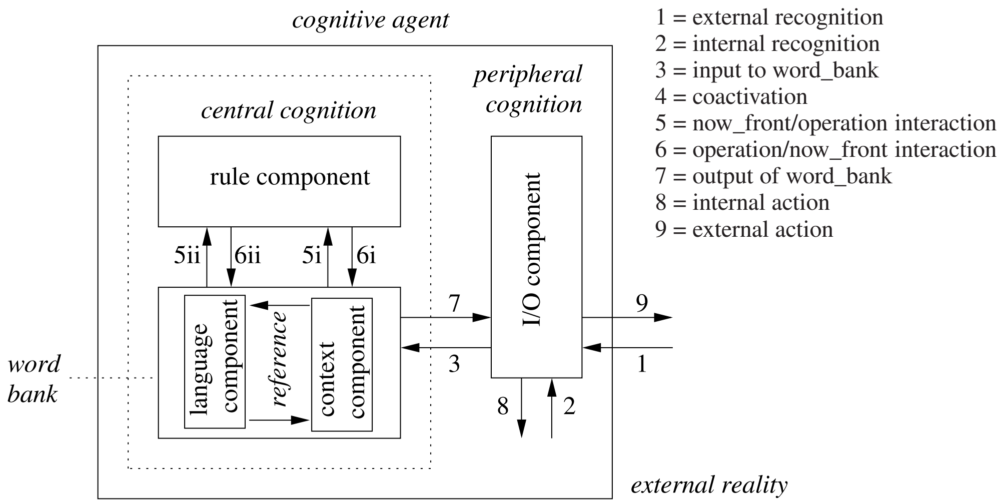
How should these different kinds of pattern matching be integrated into the component structure 4.3.2 and the functional flow of a cognitive agent as a computational mechanism?

The conceptual presentations of reference 4.3.1 and 4.3.2 have in common that reference is shown as a vertical relation between horizontal language expressions and a horizontal world. However, 4.3.2 departs from the standard assumptions of analytic philosophy, including truth-conditional semantics, because it does not treat reference as an external relation defined in a metalanguage (FoCL Chaps. 19–21), but as an agent-internal, cognitive procedure.

[12] Steels (1999) presents algorithms which automatically derive new types from similar data by abstracting from what they take to be variable in the sense of accidental. See also NLC Sect. 4.2, and FoCL Sect. 3.3.

Diagram 4.3.2 is suitable for explaining the Seven Principles of Pragmatics in the SLIM theory of language (NLC Sect. 2.6.) and for showing the applications of pattern matching based on the type-token relation (4.5.1). It fails, however, to provide a place for pattern matching based on restricted variables (4.5.2). As a solution, consider the following refined component structure, which is functionally more inclusive than diagram 4.3.2:

4.5.3 EXTENDING DIAGRAM 4.3.2

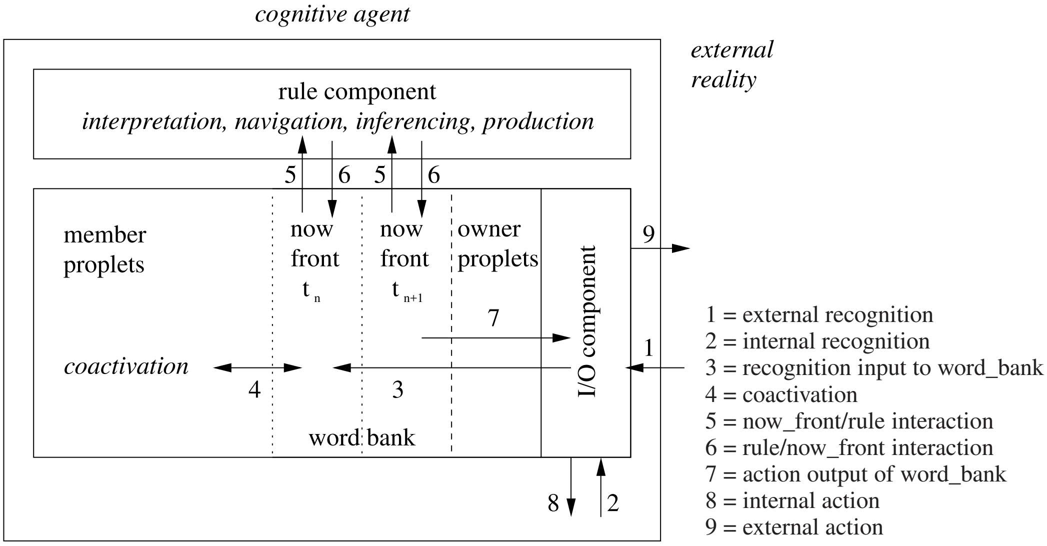


The peripheral cognition of 4.3.2 reappears as the general I/O component for external and internal recognition and action with unified input-output channels for the language and the context level. The I/O component includes automatic word form recognition (Sect. 2.5) and realization, as well as their nonlanguage counterparts (Sect. 8.1). Input to the I/O component is the raw data from external (1) and internal (2) recognition, while the internal output (3) is content written to the now front of the word bank in accordance with their core value and realized as raw data.

The component of central cognition in 4.3.2 reappears under the name *word bank*. It is rotated by 90 degrees because the token lines in a word bank are horizontal. The separation of language and context content is misleading, however, because as the storage of a proplet in a token line is determined solely by its core value and its moment of arrival, regardless of whether it is a language or a content proplet.

The technically correct way of indicating the functional flow is by abandoning the componential separation of language and context content in the database, and by showing the sorting of proplets into their token lines and in the order of their arrival instead. This may be illustrated as follows:

4.5.4 Refining the component structure 4.5.3



The numbering from 1 to 9 is the same as in 4.5.3, though without the distinctions between 5i/6i (language) and 5ii/6ii (context). Arrow 3 characterizes the now front at moment t_n as interpreting input. Arrow 7 characterizes the now front at moment t_{n+1}, passing a blueprint to the action component for realization. The right hand border of a current now front is delineated by the column of owner values; the left hand border fades into the permanent sediment of the word bank. The operations at the now front are continuously mirrored by coactivation (Sect. 5.4).

Treating proplets solely as items to be sorted into token lines according to their core values and in the order of their arrival does not abandon the distinction between language and context proplets. Instead is just one of several different *views*.[13] The view mechanism of DBS is based on connected pattern proplets:

4.5.5 Examples of different views in DBS

Different views are created by pattern proplets containing certain (kinds of) proplet values (loosely formulated):

[13] In computer science, various view mechanisms are employed for queries on demand.

1. proplets with the same prn value are viewed as a proposition,
2. proplets with NIL sur values as context content,
3. proplets with non-NIL sur values as language content,
4. proplets with interlocking pc and nc values as coordination,
5. proplet sets with successive prn values as text,
6. proplets with the same core value as a set of similar items,

etc.

Concatenated proplets are order-free (3.2.7) because all properties of a content, especially the relations between proplets, are coded by means of proplet-internal values. This is crucial for accommodating (i) the time-linear order of word form surfaces in the speak and hear modes, and (ii) the storage, retrieval, and processing of content in memory. It also provides for (iii) an unlimited number of views, as stated in MoC-3:

4.5.6 THIRD MECHANISM OF COMMUNICATION (MOC-3)

> Packaging all properties of a content proplet-internally enables a highly differentiated retrieval, based on an unlimited number of views with unlimited expressive power.

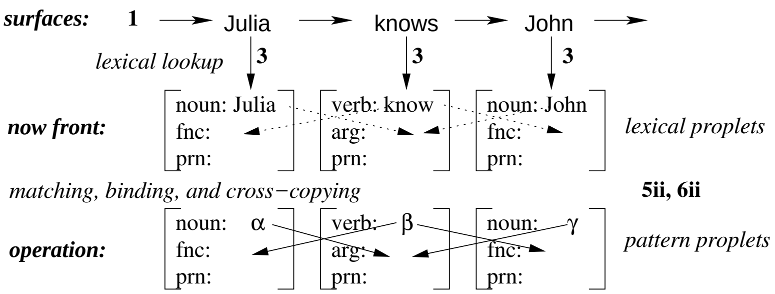
While the number of grammatical values is fixed within a given DBS system, the number of core and prn values is open (12.6.1). Therefore the number of pattern proplets using restricted variables as core and as continuation values is unlimited. The concatenation of patterns results in an unlimited number of views, including the ones shown in 3.2.2 and 3.2.5, and the ones listed in 4.5.5.

4.6 Embedding the Cycle of Communication into the Agent

The component structure 4.5.4 raises the question of how to integrate the DBS.Hear, DBS.Nav, DBS.Inf, and DBS.Speak derivations outlined in Sects. 3.3, 3.4, and 5.2, into the step by step functional flow from the agent's input to the agent's output. Furthermore, what are the impulses initiating these procedures, and where do these impulses come from?

The hear mode derivation 3.3.1 may be shown as follows (using the same numbering as in 4.5.4 to indicate corresponding inter-component mappings):

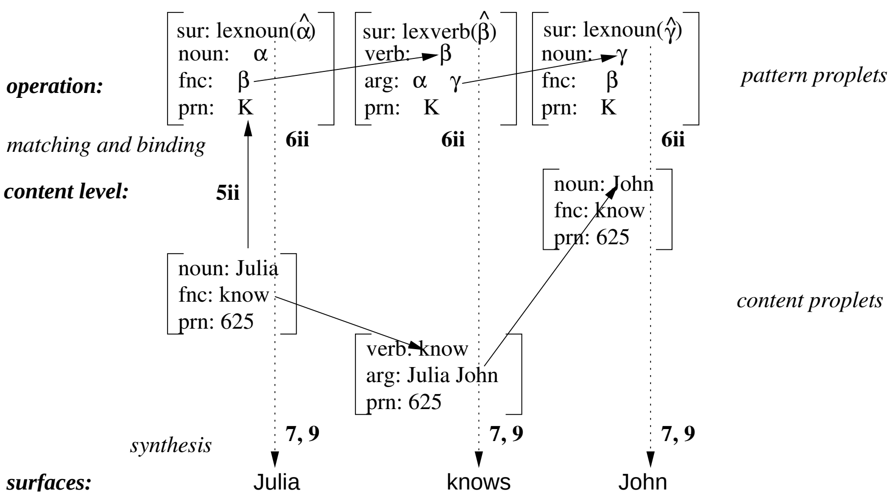
4.6.1 MAPPING INCOMING SURFACES INTO CONTENT (HEAR MODE)



The impulse activating I/O recognition (1) is a modality-dependent unanalyzed external surface. It is mapped by automatic word form recognition into a lexical proplet (3), stored at the now front according to its core value (lemmatization), and used as a sentence start. The next proplet provided by the I/O component is used as the next word. If the two proplets match the input patterns of a DBS.Hear operation (5ii), they are concatenated (6ii), thus forming a new sentence start. This process continues as long as a next word is provided.[14]

Next consider language production from stored content (3.3.3, 3.4.3) within the component structure of diagram 4.5.4:

4.6.2 MAPPING STORED CONTENT INTO SURFACES (SPEAK MODE)



Vertically, the proplets at the content level are in the alphabetical order induced by their core values. The impulse (5ii) may be provided by an urge to tell, by another agent's question (Sect. 4.2), or by a request to recount a certain

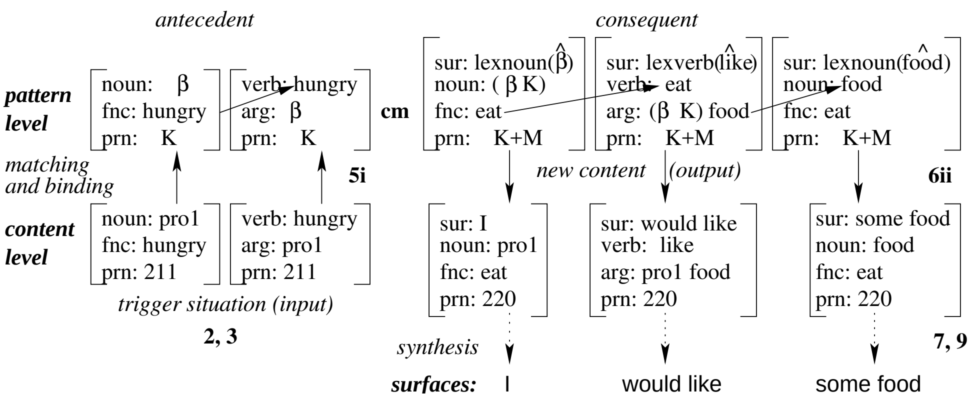
[14] For an alternative but equivalent complementary presentation see Sect. 13.3; NLC Sect. 11.3.

event.[15] In the speak mode, the operation (i) drives the navigation (diagonal arrows) and (ii) complements the proplets traversed with a sur value which is passed to one of the agent's realization components (dotted vertical arrows). Once a DBS.Nav grammar has been started by an initial impulse matching its start state, the navigation continues with an operation which matches the goal proplet resulting from the last operation with its first input pattern (self-organization).

Finally consider language production based on on-the-fly inferencing. Anticipating the discussion of inferences in Chaps. 5, 6, 10, and 11, it is sufficient for present purposes that the input and output of a DBS inference be a sequence of proplets which (i) have core and prn values and (ii) code the semantic relations of functor-argument, coordination, and coreference.

Because the content produced by on-the-fly inferencing has the same format as content resulting from recognition, it can use the same language-dependent lexicalization rules for language production (4.6.2; NLC Sects. 12.4–12.6). As an example, consider the inference β *hungry* **cm** β *eat food* (cf. Sects. 5.1, 5.3, and 5.6); the connective **cm** stands for countermeasure:

4.6.3 INFERENCE PRODUCING OUTGOING SURFACES



The impulse activating this inference is a sensation of hunger provided (2) by the agent's I/O component and passed (3) to the now front from where it triggers (5i) the antecedent. One way to use the content derived by the consequent would be as a blueprint for nonlanguage action (6i). In 4.6.3, however, the newly derived content is mapped into language (6ii), passed to the I/O component (7) and realized as a sequence of modality-dependent unanalyzed external language-dependent surfaces (9).

[15] For a general discussion of statement, question, and request dialogs see Chaps. 10 and 11.

5. Mystery Number Four: Autonomous Control

The fourth mystery is how to design autonomous control as an upscaling from the cycle of communication. The overall goal of autonomous control is to enable the agent's survival in its ecological niche. For this, the functional flow of control must connect the agent's recognition to meaningful action, including language interpretation (recognition) and production (action).

5.1 Pinball Machine Model of Cognition

For survival in the agent's ecological niche, autonomous control must maintain a continuous state of balance (equilibrium, homeostasis)[1] vis à vis a constantly changing external and internal environment. DBS uses this task as the motor driving behavior: when a deviation from the state of balance is detected, the agent's cognition tries to identify its source and to select or compute a countermeasure as best as possible.

Countermeasures are derived by relating current content to past experiences and world knowledge (automatic coactivation, Sect. 5.4). In their simplest form (FoCL Sect. 24.4), past experiences are packaged as recognition-action pairs with evaluated outcomes (appraisal). The better they fit the current task, the less the agent must rely on trial and error (Sect. 6.3). This applies also to behavior not directly related to survival: human desires for power, love, belonging, freedom, and fun may be subsumed under the balance principle by treating them as part of the agent's internal environment, like hunger.

The search for suitable content in the agent's content-addressable memory (word bank, Sect. 4.1) resembles browsing in a private library. They differ, however, in that the agent browsing in a private library is located inside the arrangement of books on their shelves, while the proplets in a word bank are located inside the cognitive agent. As a consequence, the role of the browsing user in a private library seems to be unfilled in a word bank.

[1] Bernard (1865). Wiener (1948). In more recent research, Herrmann (2003) uses the balance principle as the difference between the agent's *Ist-Zustand* (as-is-state) and *Soll-Zustand* (as-should-be-state) to trigger language production.

DBS fills this role, however, with a focus which is like a point of light navigating from one proplet to the next. Unlike with the browsing user in a private library, the control is not located in the focus. Instead, one part of controlling the motion of the focus is the semantic relations between the proplets, which are used by DBS.Nav like a railroad system for the navigation.[2] The other part is provided by actual or potential deviance from balance, which drives the focus to relate the agent's current situation to precedents in memory and derive new content serving as blueprints for action.

Putting it metaphorically, the navigation of the focus through the railroad system of connected proplets resembles a pinball machine (L&I'10), with the ball serving as the focus and the slanted playing field with its springs and levers serving as the railroad system. Because cognitive operations are agent-internal, there is no "retrieval" in the usual sense of producing an output for the user. Instead, the navigation of the focus merely *visits* successor proplets – as an activation which may be visualized as lighting up the proplets traversed, with an afterglow of suitable duration and rate of decay (annealing).

The procedure of maintaining the agent in a state of balance provides a computational reconstruction of *intention*, just as the agent's recognition and action procedures provide a computational reconstruction of *content*. By attaching certain elementary contents to language-dependent word form surfaces by means of convention, content may double as *meaning* (meaning$_1$) in DBS.

This differs from Grice (1957, 1965, 1969), who defines natural language meaning based on an undefined notion of intention:

5.1.1 Definition of meaning by Grice

> Definiendum: U meant something by uttering x.
> Definiens: For some audience A, U intends his utterance of x to produce in A some effect (response) E, by means of A's recognition of the intention.

This charming definition uses "intends" and "recognition of the intention" in the definiens without any foundation in the form of a cognitive or computational reconstruction. It plainly follows that "meant something" in the definiendum has no such foundation either.[3]

---

[2] Thus, just as the core value of a proplet serves the double function of (i) representing an important aspect of the lexical semantics and (ii) determining the location for storage and retrieval, its continuation values serve the double function of (i) representing the compositional semantics and (ii) establishing a railroad system for the focus navigation, providing each proplet with only a limited choice of successor proplets.

[3] For further discussion of Grice's notion of meaning see FoCL Chap. 4, Example II.

In DBS, the notion of intention is defined cognitively as the system-inherent (innate, instinctive) drive to maintain the agent in a state of balance.[4] The principle of balance, in turn, is realized computationally by means of inferences. An inference consists of (i) an antecedent pattern, (ii) a connective, and (iii) a consequent pattern. Examples of connectives are **caus**(e), **cm** (countermeasure), **exec**(ute), **impl**(ies), **pre**(condition), **prom**(ise), and **reas**(on).

Inferencing is used for reasoning and for deriving blueprints for action. An inference may be applied forward and backward. A forward application is triggered by content matching the antecedent (deduction, Sect. 5.2). In forward chaining, the output of the consequent of inference n must equal the input to the antecedent of inference n+1 (Continuity Condition, 5.2.1; NLC 3.6.5).

A backward application is triggered by content matching the consequent (abduction, Sects. 5.5, 13.5). In contradistinction to a forward application, a backward application does not derive a certainty (truth), but a guess at the most likely explanation. In backward chaining, the output of the antecedent of inference n must equal the input to the consequent of inference n-1.

The agent's balance may be disturbed by a nonlanguage recognition, e.g. a sensation of hunger, or a language content, e.g. a reproach or a demand (hear mode interpretation). Similarly with the countermeasure for regaining the agent's state of balance: it may be realized as a nonlanguage action, e.g. getting something to eat, or a language content, e.g. a request or an apology (speak mode production).[5]

Because all behavior, including language behavior, is managed by the agent's autonomous control, it is essential for natural language communication:

5.1.2 Fourth Mechanism of Communication (MoC-4)

> Autonomous control drives language and nonlanguage behavior to maintain a continuous state of balance vis à vis constantly changing external and internal environments. Successful behavior is defined by the agent's short-, mid-, and long-term survival in its ecological niche.

A state of balance is an absolute like truth, and like truth it provides the fix point necessary for a system of semantic interpretation. But while truth is (at least[6]) bipolar, balance is monopolar. This may be the reason why balance, unlike truth, is a dynamic principle, suitable for driving the cognitive agent

[4] The cooperative behavior of social animals, e.g. ants in a colony, may also be described in terms of balance. Such a decentralized approach to behavior is in line with Brooks (1985).

[5] See Chap. 10 for an example which illustrates the use of the balance principle for driving a dialog.

[6] See FoCL Sect. 20.5. for a discussion of non-bivalent (i.e. tripolar or multipolar) logic systems.

trying to survive, and to survive comfortably, in a constantly changing world, short-, mid-, and long-term.

Calibrating the DBS system to achieve balance in a terrain, natural or co-designed but real, requires an actual robot. After all, without the robot's external and internal interfaces we would have to recreate every nook and cranny of the changing environment by hand (as in model theory, 4.3.1). This would violate a fundamental credo of nouvelle AI, namely that The world is its own best model (Brooks 1985).

5.2 From Truth to Maintaining Balance

Inferences are operations which derive new content from given content in a meaningful way (Bibel 1993). This holds for the deductive inferences in symbolic logic,[7] which derive true conclusions (new content) from antecedents (given content). It also holds for the inferences of DBS, which are of three kinds, called R(eactor), D(eductor), and E(ffector) inferences.[8]

DBS inferences may apply in chains, based on the following principle:

5.2.1 Continuity Condition

> In the think mode, an operation AyB may only be followed by an operation BzC, if the output of AyB is matched by the antecedent of BzC, where A, B, C are patterns and y, z are connectives.

The CC provides for a simple, effective kind of self-organization (Kohonen 1988). Via identity[9] of the output of the consequent of one inference with the input to the antecedent of another, any newly added inference is automatically integrated into the agent's cognition, and similarly for backward chaining. The absence of rule packages in DBS.Hear, DBS.Nav, DBS.Speak, and DBS.Inf operations makes them structurally similar: they are all defined as sets of pattern proplets concatenated by address (Sect. 16.6).

For readability, the following example of an inference chain is simplified as follows: (i) English words are used to represent proplets, (ii) coreference by address notation is omitted, and (iii) easily programmed details regarding the iteration of values in the variable restriction (cf. line 3) are left aside.

---

[8] The terminology is intended to distinguish DBS inferences from the inferences of symbolic logic. For example, while a deductive inference like modus ponens is based on form, the reactor, deductor, and effector inferences of DBS may take content, domain, level of abstraction, etc., into account.

[9] Identity may be refined (weakened) into equivalence, for example, by using subsumption.

5.2.2 CHAINING R, D, AND E INFERENCES

1. R: β be_hungry	K	**cm**	β eat food	K+1	
2. D: β eat food	K+1	**pre**	β get food	K+2	
3. D: β get food	K+2	**down**	β get α	K+3	where $\alpha \epsilon$ {apple, pear, salad, steak}
4. E: β get α	K+3	**exec**	β locate α at γ	K+4	
5. E: β locate α at γ	K+4	**exec**	β take α	K+5	
6. E: β take α	K+5	**exec**	β eat α	K+6	
7. D: β eat α	K+6	**up**	β eat food	K+7	

Each line begins with the step number, e.g. 1, followed by the kind of inference, e.g. R(eactor), the antecedent, e.g. β be_hungry, the prn value of the antecedent, e.g. K, the connective, e.g. **cm**, the consequent, e.g. β eat food, and the prn value of the consequent, e.g. K+1. Within a chain, different occurrences of the same variable are bound to the same value (scope). The assigned value is either an address (5.3.5) or an indexical (5.2.5). In this example of forward chaining, the output of the consequent of proposition n is equal to input to the antecedent of proposition n+1 (CC, 5.2.1).

Step 1 is an R(eactor) inference (defined in 5.2.4) with the connective **cm** (countermeasure) and triggered by a sensation of hunger. Step 2 is a D(eductor) inference with the connective **pre**(condition), while step 3 is a D inference for downward traversal (defined in 6.5.9) with the connective **down**. Steps 4, 5, and 6 are E(ffector) inferences with the connective **exec**(ecute).

Step 4 may be tried iteratively for the instantiations of food provided by the consequent of step 3 (variable restriction on α). If the agent cannot locate an apple, for example, it tries next to locate a pear, and so on. Individual food preferences may be expressed by the order in the variable restriction.

Step 7 is based on a D inference for upward traversal (defined in 6.5.12) with the connective **up**. This step is called the *completor* because the consequent of the chain-final inference equals the consequent of step 1. The completor indicates the successful blueprint of a countermeasure to the imbalance characterized by the antecedent of the chain-initial reactor inference.

While R inferences are activated by triggers provided by the agent's recognition, external (e.g. *hot*) or internal (e.g. *hungry*), D and E inferences are usually initiated by other inferences which are already active. D(eductor) inferences establish content relations (Sect. 5.3), and are illustrated by synonymy (5.3.1), antonymy (5.3.2), cause and effect (5.3.3), summarizing (5.3.5), downward traversal (6.5.9), and upward traversal (6.5.12). A D inference may activate another D inference or an E inference.

E(ffector) inferences provide blueprints for the agent's action components. Because E inferences connect central cognition with peripheral cognition

[9] For a detailed discussion of modus ponens in symbolic logic and in DBS see NLC Sect. 5.3.

(4.3.2, 4.5.3, 4.5.4), their definition has to be hand in glove with the robotic hardware they are intended to operate.[10]

A limiting case of a chain is a single R/E inference, such as the following:

5.2.3 One-step chain based on an R/E inference

R/E: α feel full K **cm/exec** α stop eating K+1

Here the response to a deviation from balance results in a countermeasure which can be executed directly (i.e. without intervening D inferences).

DBS inferences resemble DBS.Nav operations like 3.4.2. However, instead of merely navigating from one proplet to the next (selective activation), an inference matches its antecedent or consequent to a content in order to derive new content. For example, using the format of pattern proplets and content proplets, the formal definition of the chain-initial R inference of 5.2.2 and its application to a content may be shown as follows:

5.2.4 Formal definition and application of a DBS inference

$$
\begin{array}{lccccccc}
 & \textit{antecedent} & & & \textit{consequent} & & \\
\begin{array}{l}\textit{pattern}\\ \textit{level:}\end{array} &
\begin{bmatrix}\text{noun: } \beta\\ \text{fnc: be\_hungry}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_hungry}\\ \text{arg: } \beta\\ \text{prn: K}\end{bmatrix} &
\textbf{cm} &
\begin{bmatrix}\text{noun: } (\beta\ \text{K})\\ \text{fnc: eat}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: eat}\\ \text{arg: } (\beta\ \text{K})\ \text{food}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{noun: food}\\ \text{fnc: eat}\\ \text{prn: K+1}\end{bmatrix} \\
 & \textit{matching and binding} & \Uparrow & & \Downarrow & & \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: be\_hungry}\\ \text{prn: 211}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_hungry}\\ \text{arg: pro1}\\ \text{prn: 211}\end{bmatrix} &
 &
\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: eat}\\ \text{prn: 211+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: eat}\\ \text{arg: pro1 food}\\ \text{prn: 211+1}\end{bmatrix} &
\begin{bmatrix}\text{noun: food}\\ \text{fnc: eat}\\ \text{prn: 211+1}\end{bmatrix} \\
 & \textit{input} & & & \textit{output} & &
\end{array}
$$

The inference is activated by the content *I am hungry* (input), which matches the antecedent pattern (forward chaining). Utilizing the values to which the variables β and K are bound (i.e. pro1 and 211, respectively), the new content *I eat food* is derived by the consequent (output).

The inclusion of the antecedent's subject in the consequent by means of the address value (β K) excludes cases in which one agent is hungry and another one eats food – which would fail as an effective countermeasure. Repeated reference by means of a 1st person indexical is exempt from the coreference by address method. Therefore the value corresponding to (β K) in the content derived by the consequent is simply pro1 rather than (pro1 211).[11]

The consequent of a DBS inference may contain variables which do not appear already in the antecedent. An example is γ in the consequent of step

---

[10] In robotics, effectors range from legs and wheels to arms and hands.

[11] For a more detailed discussion of indexicals, see Chap. 10 and Sect. 11.2.

4 in 5.2.2. This is permissible as long as the codomains of such variables are restricted to a certain range of possible values.

Which inferences are activated at any given moment is determined by the agent's current state: recognition and inferencing produce a continuous stream of proplets, which are screened and used to select corresponding content in the word bank (coactivation, Sect. 5.4). This may apply at any level of abstraction and to any choice of domain. The extent of the coactivation is determined by such parameters as the available computing power and the agent's personal interests and preferences.

The triple consisting of (i) a current or coactivated content, (ii) an inference with a matching antecedent or consequent, and (iii) the action blueprint derived is called a tia triple (trigger, inference, action). An agent's set of tia triples may be large and expanding continuously with experience.

At any given moment, the activated tia triples are evaluated in a cost-benefit manner, resulting in the necessary data reduction, i.e. a manageable number of blueprints for action. Which action is selected to be realized may be a reflex (Sect. 6.1) or a random choice (trial and error), especially when time is of the essence. But it may also be the choice of long-term planning. Technically, the final step is a switch from the subjunctive mood to the indicative mood (Sect. 5.6), resulting in external action (realization).

The application of the inference chain 5.2.2 (pattern level) to proplets in a word bank (content level) may be shown as follows:

5.2.5 New content derived by the inference chain 5.2.2

pattern level:	β	be_hungry	K	**cm**	β	eat food	K+1	**pre**
content level:	pro1	be_hungry	211		pro1	eat food	212	

β	get (food K+1)	K+2	**down**	β	get α	K+3	**exec**
pro1	get (food 212)	213		pro1	get apple	214	

β	locate α at γ	K+4	**exec**	β	take α	K+5	**exec**
pro1	locate (apple 214) at cupboard	215		pro1	take (apple 214)	216	

β	(eat K+1) α	K+6	**up**	β	(eat K+1) (food K+1)	K+7
pro1	(eat 212) (apple 214)	217		pro1	(eat 212) (food 212)	218

The four double lines should be read as one, i.e. as

pattern level:	p1	**cm**	p2	**pre**	p3	**down**	p4	**exec**	p5	**exec**	p6	**exec**	p7	**up**	p8
content level:	q1		q2		q3		q4		q5		q6		q7		q8.

The semiformal notation represents propositions by the core values of their proplets, e.g. be_hungry, and their prn value, e.g. 211. Repeated core values referring to the same item are shown as addresses, e.g. (food 212), ensuring coreference between contents throughout the chain.

The chain is activated by a trigger content, provided by the agent's I/O component. Represented as the proposition pro1 be_hungry 211, the trigger matches the antecedent of the chain-initial reactor inference (shown in more detail in 5.2.4). The new content pro1 eat food 212, derived by the consequent, serves as the trigger content for the antecedent of the second inference (step 2 in 5.2.2), and so on. The E inferences with the connective **exec** derive a sequence of blueprints for action, intended as a countermeasure.

The proplets derived by the completor consequent 211+7, i.e. pro1 (eat 212) (food 212) 218, are added to the word bank at the current end of the respective token lines (i.e. the now front, 4.1.1). Despite their prn value 218, they are equivalent to the proplets derived earlier by the consequent of the initial R inference, i.e. pro1 eat food 212, due to their definition as the indexical pro1 and the address proplets (eat 212) and (food 212).

5.3 DBS Inferences for Content Relations

The basic semantic relations of structure, i.e. classic functor-argument and coordination (Aristotle, Frege), may be used for the construction of episodic and absolute contents. One kind of absolute content is language-dependent lexical relations of meaning, such as *synonymy, antonymy, hypernymy, hyponymy, meronymy,* and *holonymy*. Another kind is language-independent, such as cause-and-effect, which we might call empirical relations. In DBS, both kinds are relations of content (i.e. independent of language surfaces).[12]

Lexical relations of meaning_1 between words are formalized as D inferences. Consider the following example of a synonymy, here between abstract and summary:

5.3.1 Inference implementing a synonymy

$$\begin{bmatrix}\text{noun: abstract}\\ \text{fnc: } \alpha\\ \text{prn: K}\end{bmatrix} \ \textbf{impl} \ \begin{bmatrix}\text{noun: summary}\\ \text{fnc: } \alpha\\ \text{prn: K+M}\end{bmatrix} \quad \text{where } \alpha \in \{\text{write, read, discuss, ...}\}$$

Accordingly, John wrote an abstract implies that John wrote a summary. The restriction on the variable α specifies likely verbs, obtained from a corpus (Sect. 15.5).

Next consider an example of a D inference implementing an antonymy, here between good and not bad:

[12] In DBS, a meaning_1 is defined as a content attached by convention to a language-dependent surface. This allows to define define the lexical relation of meaning in 5.3.1 between contents, regardless of whether or not they are attached to a language-dependent surface.

5.3.2 Inference implementing an antonymy

$$\begin{bmatrix}\text{adj: good}\\ \text{mdd: }\alpha\\ \text{prn: K}\end{bmatrix} \ \textbf{impl}\ \begin{bmatrix}\text{adj: not bad}\\ \text{mdd: }\alpha\\ \text{prn: K+M}\end{bmatrix}$$

When this inference is applied to an input content like John had a good meal, the result is the new content John had a meal which was not bad (see 6.5.9–6.5.11 for an analogous formal derivation).[13]

Thus, the lexical meaning relations defined above are formulated once as inference patterns, but the results of their application may be written many times as new content to the now front of the word bank. The new content derived by such an inference may serve to trigger further inferences, for example, for the interpretation of nonliteral uses (NLC Sect. 5.4).[14]

Parallel to the relations between contents which are attached to language surfaces (meaning$_1$) are the relations between contents which are not, most notably *cause and effect*. Due to the formal similarity between language content and context content in DBS (4.3.3), the empirical relations may be formalized in the same way as the traditional meaning relations of lexicography. As an example, consider the inference car has no fuel **caus** car does not start.

5.3.3 Inference implementing a cause and effect relation

$$\begin{bmatrix}\text{noun: car}\\ \text{fnc: have}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: have}\\ \text{arg: car no\_fuel}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: no\_fuel}\\ \text{fnc: have}\\ \text{prn: K}\end{bmatrix} \ \textbf{caus}\ \begin{bmatrix}\text{noun: (car K)}\\ \text{fnc: no\_start}\\ \text{prn: K+M}\end{bmatrix} \begin{bmatrix}\text{verb: no\_start}\\ \text{arg: (car K)}\\ \text{prn: K+M}\end{bmatrix}$$

Fuel may be instantiated by gasoline, electricity, methane, etc. (6.5.5).

The descriptive power of the DBS inferencing method may be shown further by condensing a complex content into a meaningful summary. As an example, consider the following short text, formally derived in NLC, Chaps. 13 (hear mode) and 14 (speak mode):

> The heavy old car hit a beautiful tree. The car had been speeding. A farmer gave the driver a lift.

During syntactic-semantic parsing by DBS.Hear, the content of this modest text is stored in the agent's word bank. A reasonable summary of the content would be *car accident*. This summary may be represented as follows:

---

[13] The operations 5.3.1 and 5.3.2 illustrate how traditional meaning relations may be handled in DBS. Whether or not the inverse relations hold as well is an empirical question which may be answered by distinguishing between unidirectional and bidirectional meaning inferences. Given that the inferences in question are part of an individual agent's cognition, they do not claim the status of eternal truths. Instead, they are evaluated and adapted in terms of their utility for the agent's day-to-day survival.

[14] Thus, the hypernymy inference dog **inst**(antiates) animal may derive the new content Fido is an animal from Fido is a dog. In the hear mode, this new content may be used to infer from the language expression The animal is tired that the intended utterance meaning$_2$ is Fido is tired (FoCL Sect. 4.5) – and conversely in the speak mode.

5.3.4 Relating summary car accident to text

$$
\begin{array}{llllll}
 & \multicolumn{3}{c}{\textit{member proplets}} & \textit{now front} & \textit{owner values}\\
\ldots & & & & & \\
 & & & \begin{bmatrix}\text{noun: accident}\\ \text{mdr: (car 1)}\\ \text{prn: 67}\end{bmatrix} & & \text{accident}\\
\ldots & & & & & \\
\ldots & \begin{bmatrix}\text{noun: car}\\ \text{fnc: hit}\\ \text{prn: 1}\end{bmatrix} & \begin{bmatrix}\text{noun: (car 1)}\\ \text{fnc: speed}\\ \text{prn: 2}\end{bmatrix} & \begin{bmatrix}\text{noun: (car 1)}\\ \text{mdd: accident}\\ \text{prn: 67}\end{bmatrix} & & \text{car}\\
\ldots & & & & & \\
\ldots & \begin{bmatrix}\text{verb: hit}\\ \text{arg: car tree}\\ \text{prn: 1}\end{bmatrix} & & \ldots & & \text{hit}\\
\ldots & & \begin{bmatrix}\text{verb: speed}\\ \text{arg: (car 1)}\\ \text{prn: 2}\end{bmatrix} & & & \text{speed}\\
\ldots & \begin{bmatrix}\text{noun: tree}\\ \text{fnc: hit}\\ \text{prn: 1}\end{bmatrix} & & \ldots & & \text{tree}\\
\ldots & & & & &
\end{array}
$$

The connection between propositions 1 and 2 and summary 67 is coreference by address. It is based on the original car value in proposition 1 and the corresponding address values (car 1) in proposition 2 and the summary 67. The summary consists of the *accident* proplet and a *car* address proplet. They share the prn value 67, and are connected by the modifier-modified relation, with (car 1) in in the mdr slot of *accident* and accident in the mdd slot of *(car 1)*.

How is the summary automatically derived from the text? The summary-creating inference deriving the new content with the prn value 67 is formally defined as the following D(eductor) inference, shown with the sample input and output of 5.3.4 at the word bank (content) level:

5.3.5 Summary-creating D inference

$$
\begin{array}{lllllll}
 & \textit{antecedent} & & & & \textit{consequent} & \\
\textit{pattern level:} & \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: hit}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: hit}\\ \text{arg: }\alpha\ \beta\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{noun: }\beta\\ \text{fnc: hit}\\ \text{prn: K}\end{bmatrix} & \textbf{sum} & \begin{bmatrix}\text{noun: }(\alpha\text{ K})\\ \text{mdd: accident}\\ \text{prn: K+M}\end{bmatrix} & \begin{bmatrix}\text{noun: accident}\\ \text{mdr: }(\alpha\text{ K})\\ \text{prn: K+M}\end{bmatrix}\\
 & \multicolumn{6}{l}{\text{where } \alpha \in \{\text{car, truck, boat, ship, plane, ...}\} \text{ and } \beta \in \{\text{tree, rock, wall, mountain, ...}\} \cup \alpha}\\
 & \textit{matching and binding} & \Uparrow & & & \Downarrow & \\
\textit{content level:} & \begin{bmatrix}\text{noun: car}\\ \text{fnc: hit}\\ \text{prn: 1}\end{bmatrix} & \begin{bmatrix}\text{verb: hit}\\ \text{arg: car tree}\\ \text{prn: 1}\end{bmatrix} & \begin{bmatrix}\text{noun: tree}\\ \text{fnc: hit}\\ \text{prn: 1}\end{bmatrix} & & \begin{bmatrix}\text{noun: (car 1)}\\ \text{mdd: accident}\\ \text{prn: 67}\end{bmatrix} & \begin{bmatrix}\text{noun: accident}\\ \text{mdr: (car 1)}\\ \text{prn: 67}\end{bmatrix}
\end{array}
$$

Antecedent and consequent are related by the connective **sum**(marize).

In the inference, the possible values which α and β may be bound to during matching are restricted by the codomains of these variables: the restricted variable α generalizes the summary-creating inference to different kinds of accidents, e.g. *car accident, truck accident,* etc., while the restricted variable β limits the objects to be hit to trees, rocks, etc., as well as cars, trucks, etc. Any content represented by the proplet *hit* with a subject and an object proplet satisfying the variable restrictions of α and β, respectively, will be automatically (i) summarized as an accident of a certain kind where (ii) the summary is related to the summarized by means of an address value, here (car 1), thus fulfilling the condition that the data in a word bank may not be modified (Sect. 14.5).

By summarizing content into shorter and shorter versions, there emerges a hierarchy providing retrieval relations for upward or downward traversal (Sect. 6.5). Upward traversal supplies the agent with more general notions at a higher level of abstraction, while downward traversal supplies more concrete instantiations at a lower level of abstraction. Either kind may be used to access and to apply inferences defined at another level of abstraction, and to subsequently return to the original level.

5.4 Automatic Coactivation

The amount of data in the agent's memory may be very large. For efficiently finding contents in memory which match the agent's current situation DBS introduces a new database mechanism called *coactivation*. It is a kind of guided association which continuously accompanies the agent's current cognition with corresponding content stored in the word bank.

Coactivation works like a dragnet, pulled by the concepts activated by the agent's current recognition, inferencing, and action. As a form of association,[15] coactivation results in a mild form of selective attention which accompanies the agent's current reasoning with relevant knowledge and past experiences.

A coactivation consists of three steps and may be primary or secondary, the latter with an open number of degrees n≥0. The first step is the *subactivation* of the token line which corresponds to a trigger concept provided by the agent's current situation. Intuitively, a subactivation may be viewed as highlighting an area of content at half-strength, setting it off against the rest of the word bank, but such that exceptional evaluations are still visible as brighter spots. In this way, the agent will be alerted to potential threats or opportunities even in current situations which would otherwise seem innocuous.

Consider the following example:

[15] Such as associating a happy memory with a certain place.

5.4.1 Trigger concept subactivating corresponding token line

member proplets					*now front*	*owner value*		*trigger concept*
adj: hot mdd: potato prn: 20	adj: hot mdd: water prn: 32	adj: hot mdd: potato prn: 55	adj: hot mdd: day prn: 79	. . .		hot	⇐	*hot*

The trigger concept *hot* is supplied by the agent's external recognition and matches the owner value hot in the agent's word bank. This subactivates the associated token line.

However, if a continuous sequence of trigger concepts were to always subactivate complete token lines, the resulting amount of data would be too large to be useful. Therefore, the second step of a coactivation is to use the classic semantic relations of functor-argument and coordination connecting the incoming concepts for the *intersection* of the relevant token lines. In this way, the more semantically connected the concepts coming in, the more narrow and specific the coactivated data (search space reduction).

For example, if the agent's current recognition relates *hot* and *potato* as modifier-modified, the token lines of *hot* and *potato* might contain the following intersections, indicated typographically by bold face:

5.4.2 Intersecting token lines for *hot* and *potato*

	member proplets				*now front*	*owner values*
. . .	**adj: hot** **mdd: potato** **prn: 20**	adj: hot mdd: water prn: 32	**adj: hot** **mdd: potato** **prn: 55**	adj:hot mdd: day prn: 79		hot
. . .						
. . .	**noun: potato** **fnc: look_for** **mdr: hot** **prn: 20**	noun: potato fnc: cook mdr: big prn: 35	**noun: potato** **fnc: find** **mdr: hot** **prn: 55**	noun: potato fnc: eat mdd: small prn: 88		potato

The example contains two intersections, each consisting of two proplets sharing (i) a prn value and (ii) the modifier-modified relation between *hot* and *potato*. The intersections differ from each other in their respective prn values, 20 and 55, and the fnc values of the nouns, look_for and find.

The third step of a primary coactivation is *completion*. Using DBS.Nav, an intersection is completed into a full proposition by navigating along the intrapropositional semantic relations (spreading activation, Quillian 1968), activating the proplets traversed at half strength, like subactivation. For example, using the functor-argument coded by the leftmost proplets in 5.4.2, DBS.Nav may complete the intersection into a proposition (primary completion):

5.4.3 Completion of an intersection by spreading activation

noun: John fnc: look_for prn: 20	verb: look_for arg: John potato nc: (serve 21) prn: 20	**noun: potato** **fnc: look_for** **mdr: hot** **prn: 20**	**adj: hot** **mdd: potato** **prn: 20**

The completion is based on the addresses (potato 20) in the *hot* proplet, (look_for 20) in the *potato* proplet, and (John 20) in the *look_for* proplet.

While a primary completion utilizes the intrapropositional relations of functor-argument and coordination (NLC Chaps. 6 and 8), a secondary completion is based on the corresponding extrapropositional relations (NLC Chaps. 7 and 9). For example, using the nc value of the *look_for* proplet in 5.4.3, a secondary completion of degree 2 may spread from John looked for a hot potato to the successor propositions with the verb value serve and the prn value 21. [16]

The degree of a secondary coactivation corresponds to the degree of its completion. The degree is automatically selected at any current moment, including no secondary coactivation at all. It depends on the computational resources available as well as the agent's interests and current preoccupation. Compare, for example, running for one's life and a leisurely walk, both in the same park: the very same triggers will be completely ignored in the first case, but may result in rich coactivations in the second.

5.5 Analogical Models for Problem Solving

Sect. 5.2 showed a *data-driven*[17] application of the inference chain 5.2.2 (forward chaining). Let us consider now a *goal-driven*[18] application of this same inference chain (backward chaining).[19] While a data-driven application is for instinctive, habitual, and rote behavior, a goal-driven application is for nonroutine behavior such as analogical reasoning and problem solving. The backward-chaining of a goal-driven inference application makes heavy use of coactivation.

[16] Thus, the movie title All the President's Men (Pakula 1976) will likely activate Couldn't put Humpty Dumpty Together Again as a continuation, referring to R. M. Nixon. In fiction, our notion of triggering a spreading coactivation is illustrated by the madeleine experience of Proust (1913), which brings back an almost forgotten area from what he calls "l'édifice immense du souvenir."

[17] So-called because the inference chain is triggered by data matching the initial antecedent.

[18] So-called because the inference chain is triggered by a goal matching a consequent. A goal-driven application is a form of case-based reasoning (Schank 1982). See also Muñoz-Avila et al. (2010).

[19] Our use of the terms *data-driven* and *goal-driven* follows the standard practice in computer science, even though from the viewpoint of DBS the input to goal-driven inferencing, i.e. the goal, is data too – no different from that triggering data-driven inferencing. A more adequate terminology would be antecedent-driven for data-driven and consequent-driven for goal-driven.

For example, the observation of another agent eating an apple matches the consequent K+6 of the penultimate inference in 5.2.2. This observation may set in motion a search to determine where the apple came from in order get one for oneself. Like forward chaining, backward chaining fulfills the Continuity Condition (CC) 5.2.1.

The search starts with intersections between (i) the agent's current task and (ii) potential countermeasures stored in the agent's memory, observed or self-performed. Of the available countermeasures, the one (a) best matching the agent's current situation and (b) with the best outcome is automatically selected. If no such countermeasure is available, the agent's options are some additional inferencing and ultimately trial and error.

If an appropriate countermeasure has been found, the second step is a *transfer* of the content in question. The transfer replaces the agent of the remembered countermeasure by pro1, and provides new prn and address values. The result is a new content, written to the now front and serving as a blueprint for an action sequence suitable for reestablishing the agent's balance.

Assume, for example, that the agent is alone in Mary's house – which serves as a trigger (5.4.1) subactivating the token line of *Mary* in the agent's word bank. Furthermore, the agent is hungry, which triggers the *hungry-eat* inference chain 5.2.2. The constant *eat* in the consequent of the completor inference of the chain subactivates the corresponding token line, resulting in intersections between the *Mary* and the *eat* token lines such as the following:

5.5.1 Two *Mary eat* intersections

$$\begin{bmatrix}\text{noun: (Mary 25)}\\ \text{fnc: eat}\\ \text{prn: 48}\end{bmatrix} \qquad \begin{bmatrix}\text{noun: (Mary 25)}\\ \text{fnc: eat}\\ \text{prn: 82}\end{bmatrix}$$

$$\begin{bmatrix}\text{verb: eat}\\ \text{arg: (Mary 25) } \textbf{(apple 46)}\\ \text{prn: 48}\end{bmatrix} \qquad \begin{bmatrix}\text{verb: eat}\\ \text{arg: (Mary 25) (müsli 80)}\\ \text{prn: 82}\end{bmatrix}$$

In other words, the agent remembers Mary in her house eating an apple and eating müsli.

The two proplets in each intersection share a prn value, namely 48 and 82, respectively, and are in two semantic relations of structure, namely subject/predicate and object/predicate. In both intersections, the verb proplet *eat* provides at least one yet unrealized intrapropositional continuation, namely (apple 46) 48 in the first and (müsli 80) 82 in the second. Following the continuation in the first intersection results in the following primary (5.4.3) completion:

5.5.2 Completion spreading from *Mary eat* to *Mary eat apple*

$$\begin{bmatrix}\text{noun: (Mary 25)}\\ \text{fnc: eat}\\ \text{prn: 48}\end{bmatrix} \begin{bmatrix}\text{verb: eat}\\ \text{arg: (Mary 25) } \textbf{(apple 46)}\\ \text{prn: 48}\end{bmatrix} \begin{bmatrix}\text{noun: (apple 46)}\\ \text{fnc: eat}\\ \textbf{eval: attract}\\ \text{prn: 48}\end{bmatrix}$$

The *(apple 46) 48* proplet contains the feature [eval: attract]. Assuming that the corresponding subactivation for the second intersection in 5.5.1 happens to evaluate the *(müsli 80) 82* proplet as [eval: avoid][20] (not shown), the agent will use only the first, and not the second, subactivation as a trigger situation to activate a consequent in the inference chain 5.2.2:

5.5.3 Stored content matching consequent in inference chain

```
pattern level:  β be_hungry  K cm  β eat food K+1    pre
content level:       #               #

          β get food  K+2  down  β get α K+3    exec
                #                  #

          β locate α  at γ  K+4  exec  β take α  K+5    exec
                #                         #

          β            eat  α            K+6  up  β eat food  K+7
          (Mary 25) eat  (apple 46)  48                 #
```

The trigger content (Mary 25) eat (apple 46) 48 matches the consequent K+6 of the penultimate inference. All other inference parts in the chain do not have matching contents at this point, indicated by #.

Next, the matching content (Mary 25) eat (apple 46) 48 is used for a secondary (i.e. extrapropositional) completion. It includes propositions which precede at the agent's content level, and which contain nouns coreferent with (Mary 25) and with (apple 46). Pattern matching selects coactivated propositions which fit antecedents and consequents preceding β eat α K+6 in the inference chain 5.2.2, resulting, for example, in the following correlation of inferences and contents:

5.5.4 Extending content by secondary subactivation

```
pattern level:  β be_hungry  K cm  β eat food K+1    pre
content level:       #               #

          β get food  K+2 down  β get α K+3    exec
                #                 #
```

[20] The assumed evaluations reflect the agent's personal preference for eating apples over eating müsli.

β	locate	α	at γ	K+4	**exec**
(Mary 25)	locate	apple	at cupboard	46	

β	take	α	K+5	**exec**
(Mary 25)	take	(apple 46)	47	

β	eat	α	K+6	**up**	β eat food	K+7
(Mary 25)	eat	(apple 46)	48		#	

At this point, the chain with partially[21] matching contents may be used by the hungry agent as a model for regaining balance. All that is required is to replace Mary by pro1 (transfer) and to derive new content with new prn and address values as blueprints for action:

5.5.5 Transfer and completion

pattern level:	β be_hungry	K	**cm**	β eat food	K+1	**pre**
content level:	#			#		

β get food	K+2	**down**	β get α	K+3	**exec**
#			#		

β	locate	α	at γ	K+4	**exec**
pro1	locate	apple	at cupboard	91	

β	take	α	K+5	**exec**
pro1	take	(apple 91)	92	

β	eat	α	K+6	**up**	β eat food	K+7
pro1	eat	(apple 91)	93		pro1 eat food	94

The content propositions following the **exec** connective at the pattern level, i.e. 91–93, are blueprints for action.[22] They are ready for realization, but the final decision about whether or not they are actually passed on to the agent's action components (4.5.4, interface 7) is still open.

5.6 Subjunctive Transfer

The agent's decision of whether or not to realize a current blueprint for action is implemented as a change of what is called the *verbal mood* in linguistics. In English, the verbal moods are the indicative, the subjunctive, and the imperative (NLC, A.3.2, 8). In DBS, they are formally implemented as the ind, sbjv, and impv values of the sem(antics) attribute in verb proplets (3.5.6).

[21] If the agent were to assume (unnecessarily) that Mary must have been hungry (thus supplying content all the way to the initial R inference), then this would correspond to an abductive inference in logic. For the purpose at hand, observing the occasion of Mary locating, taking, and eating an apple is sufficient.

[22] If the propositions 91–93 have been executed successfully, proposition 94 completes the countermeasure for regaining balance.

Content derived by the agent’s recognition is indicative. A blueprint derived by inferencing, but not yet passed to the agent’s action components, is subjunctive. Old content may be of any verbal mood and does not initiate any action directly. However, activated old content may initiate the derivation of new content by matching the initial antecedent (forward chaining) or the consequent (backward chaining) of an inference. The propositions of this new content may also be of any verbal mood, but only imperative blueprints following the **exec** connective are passed to the agent’s action components for realization.

A subjunctive blueprint may be revised into an imperative blueprint by means of the following inference, which (i) copies the content matching the antecedent into the consequent, (ii) changes the verbal mood value from subjunctive to imperative, and (iii) uses the connective **mc**, for *mood change*:

5.6.1 Inference changing subjunctive to imperative content

$$\begin{bmatrix}\text{verb: } \alpha \\ \text{sem: X } \textbf{sbjv} \text{ Y} \\ \text{prn: K}\end{bmatrix} \quad \textbf{mc} \quad \begin{bmatrix}\text{verb: } \alpha \\ \text{sem: X } \textbf{impv} \text{ Y} \\ \text{prn: K+M}\end{bmatrix}$$

$$\begin{bmatrix}\text{noun: pro1} \\ \text{fnc: take} \\ \text{prn: 92}\end{bmatrix} \begin{bmatrix}\text{verb: take} \\ \text{arg: (apple 91)} \\ \text{sem: } \textbf{sbjv} \\ \text{prn: 92}\end{bmatrix} \begin{bmatrix}\text{noun: (a. 91)} \\ \text{fnc: take} \\ \text{prn: 92}\end{bmatrix} \qquad \begin{bmatrix}\text{noun: pro1} \\ \text{fnc: take} \\ \text{prn: 95}\end{bmatrix} \begin{bmatrix}\text{verb: take} \\ \text{arg: (apple 91)} \\ \text{sem: } \textbf{impv} \\ \text{prn: 95}\end{bmatrix} \begin{bmatrix}\text{noun: (a. 91)} \\ \text{fnc: take} \\ \text{prn: 95}\end{bmatrix}$$

The content matching the antecedent is the same as the proposition 92 in 5.5.5 except that it is represented explicitly by proplets. Whether or not the **mc** inference is applied and the action sequence is actually attempted depends on agent-internal and -external circumstances. An internal circumstance leading to the application would be a high need for a countermeasure, while a low need may leave the application unexecuted (laziness). An external circumstance stopping the application may be an override, such as a ringing phone.

If the **mc** inference is applied, the success of the attempted action sequence will depend on whether the agent’s environment provides the necessary preconditions. For example, the countermeasure derived in 5.5.5 will be successful only if proposition 91 turns out to hold in the agent’s current situation.

In addition to the switching of a subjunctive blueprint into an imperative one, there is also the switching of an indicative content into the subjunctive, especially in combination with transfer. An example of such a *subjunctive transfer* is *empathy*, i.e. an agent’s ability to share another being’s emotions and feelings. For example, when agent A observes in a movie how actor B is being attacked by a monster, A may empathize with B by subjunctive transfer, i.e. by tentatively replacing B with A in the content transported by the movie (as in the transfer from Mary in 5.5.4 to pro1 in 5.5.5). In this way, A may experience the agony of B while knowing that there is no danger – experienced as a pleasant thrill.[23] This is easily modeled in an artificial agent.

Subjunctive transfer is also essential for constructing the agent's *discourse model* in dialog (Dominey and Warneken 2011). Consider, for example, two agents discussing going on a vacation together. Their respective discourse models include not only knowledge of the subject matter, e.g. where to go, how to get there, where to stay, what to do there, how to pay for it, etc., but also knowledge of the other agent. While knowledge of the subject matter is subactivated by means of intersection, knowledge of the other agent's cognitive state is based also on subjunctive transfer.

Even though subjunctive transfer will provide a maximum of information about another agent's viewpoints, feelings, tolerances, preferences, dislikes, demands, etc., it may not be enough. For example, agent A may inadvertently hit a sensitive spot of B, hitherto unknown and with explosive consequences. Thus, goal-driven behavior is much less predictable than data-driven behavior – though both may use the same inference chain, e.g. 5.2.2.

In summary, instinctive, habitual, or rote behavior provided by data-driven inferencing is the same over and over again and is therefore predictable in another agent. Another agent's goal-driven behavior,[24] in contrast, is unpredictable because it depends on data stored in the other agent's memory and is therefore not directly accessible to the partner in discourse.

The DBS inferencing described is continuously triggered by (i) current recognition and (ii) coactivation (Sects. 5.4, 14.3). The result is a stream of new inference data written to the now front – in addition to current language and nonlanguage recognition. This raises the more general question of how to avoid overflow of the agents' large, but finite memory (Sect. 14.6).

---

[23] A related phenomenon is identifying with the fortunes of some favorite sports team.

[24] Goal-driven behavior is important to nouvelle AI in general (Braitenberg 1984, Brooks 1991) and BEAM robotics (Biology, Electronics, Aesthetics, Mechanics) in particular (Tilden and Hasslacher 1996). However, while BEAM has the goal to model goal-driven behavior with the simplest means possible, based on analogical sensors without microprocessors, DBS uses representations of content in a database. This is because DBS starts out from natural language, while BEAM robotics proceeds from insect behavior.

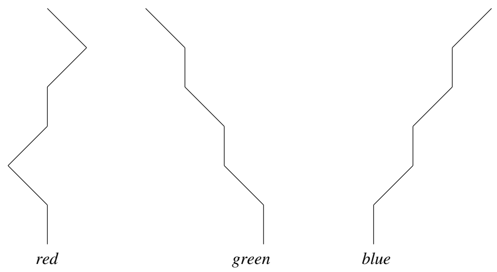
6. Mystery Number Five: **Learning**

The wide variety of DBS grammar operations and DBS inferences described so far would be of limited use if they had all to be defined and adjusted by hand. Therefore, the fifth mystery of natural communication is how to model (i) *adaptation* during evolution in phylogenesis and ontogenesis, including language acquisition, and (ii) *learning* as an improvement of the agent's survival skills in a changing environment.

6.1 Fixed Behavior Agents

The phylogenetic and ontogenetic evolution of an agent's cognition begins with fixed behavior, based on the fixed action patterns (FAPs) of ethology (Campbell 1996). As an abstract example consider an agent which can perceive no more than three external stimuli, namely a red, a green, and a blue light (recognition), and can perform no more than three kinds of external motion, namely straight, left, and right (action). Furthermore, a red light triggers a straight, left, right, straight, right, left sequence, a green light triggers a straight, left, straight, left, straight, left sequence, and a blue light triggers a straight, right, straight, right, straight, right sequence. Without going into the details of angle, length, etc., of each kind of step, a graphical representation of these action sequences may be shown roughly as follows:

6.1.1 Motion patterns of a fixed behavior agent



Each of these fixed stimulus-response pairs may be utilized for (i) conspecific communication, e.g. as a mating dance or a greeting, or for (ii) non-communicative behavior, e.g. hiding by wiggling into the sand or as a pattern of flight.

In adaptive behavior (Sect. 5.5), recognition action events are recorded in the temporal order of their arrival in the word bank. This is indicated by using natural numbers as prn values, as in the extrapropositional coordination 3.2.4.

In fixed behavior, in contrast, recognition action sequences are used as patterns. When triggered, they are repeated without keeping track of individual instances.[1] This is indicated by using prn values consisting of two parts, a letter common to all proplets of the sequence and subscripted natural numbers specifying the order within the sequence. Consider the following set of proplets, which codes the first behavior pattern in 6.1.1:

6.1.2 Coding motion triggered by red as a set of proplets

$$\begin{bmatrix}\text{rec: red}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: } x_1\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv: red}\\ \text{nxt: left}\\ \text{prn: } x_2\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv: strght}\\ \text{nxt: right}\\ \text{prn: } x_3\end{bmatrix} \begin{bmatrix}\text{act: right}\\ \text{prv: left}\\ \text{nxt: strght}\\ \text{prn: } x_4\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv: right}\\ \text{nxt: right}\\ \text{prn: } x_5\end{bmatrix} \begin{bmatrix}\text{act: right}\\ \text{prv: strght}\\ \text{nxt: left}\\ \text{prn: } x_6\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv: right}\\ \text{nxt:}\\ \text{prn: } x_7\end{bmatrix}$$

The sequence begins with a *red* proplet, which represents the stimulus activating the motion pattern. The next attribute of the *red* proplet has the value strght, the next attribute of the *strght* proplet has the value left, etc.

Similarly for the other two motion patterns of 6.1.1:

6.1.3 Coding motion triggered by green as a set of proplets

$$\begin{bmatrix}\text{rec: green}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: } y_1\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv: green}\\ \text{nxt: left}\\ \text{prn: } y_2\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv: strght}\\ \text{nxt: strght}\\ \text{prn: } y_3\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv: left}\\ \text{nxt: left}\\ \text{prn: } y_4\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv: strght}\\ \text{nxt: strght}\\ \text{prn: } y_5\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv: left}\\ \text{nxt: left}\\ \text{prn: } y_6\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv: strght}\\ \text{nxt:}\\ \text{prn: } y_7\end{bmatrix}$$

6.1.4 Coding motion triggered by blue as a set of proplets

$$\begin{bmatrix}\text{rec: blue}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: } z_1\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv: blue}\\ \text{nxt: right}\\ \text{prn: } z_2\end{bmatrix} \begin{bmatrix}\text{act: right}\\ \text{prv: strght}\\ \text{nxt: strght}\\ \text{prn: } z_3\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv: right}\\ \text{nxt: right}\\ \text{prn: } z_4\end{bmatrix} \begin{bmatrix}\text{act: right}\\ \text{prv: strght}\\ \text{nxt: strght}\\ \text{prn: } z_5\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv: right}\\ \text{nxt: strght}\\ \text{prn: } z_6\end{bmatrix} \begin{bmatrix}\text{act: right}\\ \text{prv: strght}\\ \text{nxt:}\\ \text{prn: } z_7\end{bmatrix}$$

The storage and retrieval of these sequences in a word bank is content-addressable, but their activation is not for deriving new content.

Therefore, fixed behavior uses only a fixed amount of memory, in contradistinction to adaptive behavior. This means that a *non-writable* memory is sufficient for modeling fixed behavior.

---

[1] Webb et al. (2009) classify increasingly powerful software systems of behavior in their Fig. 1.2.

The sequences 6.1.2–6.1.4 are performed by LA.act. Because FAP agents have no writable memory (Sect. 6.3), LA.act uses rule packages, like the formal LA grammars in FoCL, Part II. The definition begins with the variables:

6.1.5 Variable definition of LA.act

T_n ϵ {red, green, blue} and n ϵ {1, 2, 3, ...}
M1 ϵ {strght, left, right}
M2 ϵ {strght, left, right}
K ϵ {x_i, y_i, z_i, ...} and i ϵ {1, 2, 3, ...}

T stands for trigger and M for motion. The variables are restricted to the values in the corresponding sets. The rule system of LA.act is defined as follows:

6.1.6 Rule system of LA.act

$\mathbf{ST}_S =_{def}$ { ($\begin{bmatrix}\text{rec: T}_n\end{bmatrix}$ {Rule_0, Rule_1}) }

Rule_0 {Rule_0, Rule_1}

$$\begin{bmatrix}\text{rec: T}_n\\ \text{nxt: T}_{n+1}\\ \text{prn: K}_i\end{bmatrix} \Rightarrow \begin{bmatrix}\text{rec: T}_{n+1}\\ \text{prv: T}_n\\ \text{prn: K}_{i+1}\end{bmatrix} \qquad \text{output position nw}$$

Rule_1 {Rule_2}

$$\begin{bmatrix}\text{rec: T}_n\\ \text{nxt: M1}\\ \text{prn: K}_i\end{bmatrix} \Rightarrow \begin{bmatrix}\text{act: M1}\\ \text{prv: T}_n\\ \text{prn: K}_{i+1}\end{bmatrix} \qquad \text{output position nw}$$

Rule_2 {Rule_2}

$$\begin{bmatrix}\text{act: M1}\\ \text{nxt: M2}\\ \text{prn: K}_i\end{bmatrix} \Rightarrow \begin{bmatrix}\text{act: M2}\\ \text{prv: M1}\\ \text{prn: K}_{i+1}\end{bmatrix} \qquad \text{output position nw}$$

$\mathbf{ST}_F =_{def}$ {($\begin{bmatrix}\text{nxt: }\end{bmatrix}$ $\text{rp}_{\text{Rule\_2}}$)}

LA.act automatically starts navigating through the agent's non-writable word bank whenever its start state ST_S is satisfied. This is the case whenever the agent's recognition provides one of the values of the trigger variable T_n (6.1.5), i.e. whenever the agent perceives a red, green, or blue light.

The rule package of the start state ST_S calls Rule_0 and Rule_1. Rule_0 may be applied repeatedly to parse complex stimuli, like red green or red red red. Rule_1 moves from the end of the stimulus to the beginning of the motion sequence and calls Rule_2. Rule_2 calls itself and completes the motion sequence. After that the fixed behavior agent comes to rest until it is triggered by another stimulus.

As an example of a rule application consider Rule_1, called from the start state (rather than from Rule_0) and applied to the input proplet *red*:

6.1.7 Applying Rule_1 of LA.act to a red trigger

$$
\begin{array}{lllll}
 & \textbf{Rule\_1} \quad \{\text{Rule\_2}\} & & & \\
\textit{pattern level:} & \begin{bmatrix}\text{rec: } T_n\\ \text{nxt: M1}\\ \text{prn: } K_i\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{act: M1}\\ \text{prv: } T_n\\ \text{prn: } K_{i+1}\end{bmatrix} & \text{output position nw}\\
\textit{matching and binding} \ \Uparrow & & & \Downarrow & \\
\textit{content level:} & \begin{bmatrix}\text{rec: red}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: } x_1\end{bmatrix} & & \begin{bmatrix}\text{act: strght}\\ \text{prv: red}\\ \text{nxt: left}\\ \text{prn: } x_2\end{bmatrix} &
\end{array}
$$

By matching the first pattern proplet to the corresponding first proplet at the content level, the trigger variable T_n is bound to red, the motion variable M1 is bound to strght, and the prn variable K_i is bound to x_1. In this way, the variables in the second pattern proplet are provided with values which are sufficient to retrieve (navigate to, activate) the second proplet shown at the content level,[2] i.e. the proplet with the core value strght and the prn value x_2.

After the successful application of Rule_1, its rule package calls Rule_2, which takes the second proplet of 6.1.7 (word bank level) as input and continues the navigation through the stimulus-response sequence in question:

6.1.8 Applying Rule_2 of LA.act to a strght motion

$$
\begin{array}{lllll}
 & \textbf{Rule\_2} \quad \{\text{Rule\_2}\} & & & \\
\textit{pattern level:} & \begin{bmatrix}\text{act: M1}\\ \text{nxt: M2}\\ \text{prn: } K_i\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{act: M2}\\ \text{prv: M1}\\ \text{prn: } K_{i+1}\end{bmatrix} & \text{output position nw}\\
\textit{matching and binding} \ \Uparrow & & & \Downarrow & \\
\textit{content level:} & \begin{bmatrix}\text{act: strght}\\ \text{prv: red}\\ \text{nxt: left}\\ \text{prn: } x_2\end{bmatrix} & & \begin{bmatrix}\text{act: left}\\ \text{prv: strght}\\ \text{nxt: right}\\ \text{prn: } x_3\end{bmatrix} &
\end{array}
$$

Again, the variables of the first pattern proplet are bound to the corresponding values of the corresponding proplet at the content level (which is the proplet the previous rule application 6.1.7 navigated to). With these values, the second pattern proplet retrieves the second proplet shown at the content level, i.e. the motion proplet *left*. Because the rule package of Rule_2 calls Rule_2, this rule is reapplied until there are no more sequence-internal successor proplets in the word bank.

[2] Except for the presence of a rule package, the mechanism of this navigation is the same as the one illustrated in 3.4.2 for DBS.Hear and in 3.4.3 for DBS.Nav.

The final proplet of a motion sequence is characterized formally by its empty next value, just as the initial proplet has an empty prev value (6.1.2, 6.1.3, 6.1.4, 6.1.7). The word bank proplets activated by the navigation of LA.act are used as blueprints for the agent's action components.

LA.act is simple because the concatenation between proplets is limited to the continuation attributes prev and next. Also, building the hardware for the core values *red, green, blue, strght, left,* and *right* should not be too difficult. The LA.act system illustrates the basic DBS constructs listed in 1.2.2.

LA.act resembles the propositional calculus of symbolic logic in that both model *coordination*.[3] They differ in that propositional calculus is designed to define the truth conditions of formulas like $(p \wedge q) \vee r$, while LA.act is designed to model the behavior of an agent in the form of stimulus-response sequences. Accordingly, the constants and variables of standard bivalent propositional calculus have only two semantic values, namely true (also written as T, 1, or $\top$) and false (also written as F, 0, or $\perp$), while the semantic values of LA.act proplets comprise an open number of semantic values, like those defined 6.1.5, resulting in an open number of recognition and action procedures, like those defined in 6.1.6.

6.2 Guided Patterns to Expand a Fixed Behavior Repertoire

The interaction between a non-writable word bank and LA.act may be compared to the interaction between a record disc and a record player.[4] However, while the time-linear sequence of signals is stored on a disc in one long record groove, the database schema of a word bank is organized in alphabetical token lines containing proplets which are accessed according to their core and prn value, whereby the current proplet specifies its successor by address.

Providing a fixed behavior agent with additional stimulus-response pairs requires translating them into sets of proplets (as in 6.1.2–6.1.4), which are stored in the agent's word bank. Thus, the burden of handling additional motion patterns does not require an extension of the LA.act rule system (record player), but an extension of the data stored in the word bank (record disks).[5]

---

[3] See Hausser 2003 for a reconstruction of propositional calculus in DBS, with special attention to Boolean Satisfiability (SAT).

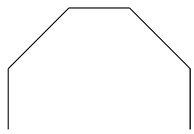
[4] This two-level structure is one of several differences between the formalism of LA grammar and formalisms based on possible substitutions such as phrase structure grammar, including finite state automata. For a comparative complexity analysis of LA grammar and PS-grammar, see FoCL Chaps. 8–12. For an analysis of the relation between LA grammar and FSAs, see CoL Sect. 8.2.

[5] An alternative model for controlling behavior is the *goal stacks* or *goal trees* in the framework of Anderson's ACT-* theory (Corbett et al. 1988).

The addition of new stimulus-response pairs may be programmed directly into the word bank, but it may also be "taught" to a fixed behavior agent by showing it the stimulus and guiding it through the response – like a tennis coach taking a player's arm and guiding it through a particular strike motion.[6]

As a formal example of what we call the *guided pattern* method in DBS, consider providing the fixed behavior agent described in the previous section with the new stimulus sequence red green and guiding it through the new motion sequence strght, left, left, left, left, as shown graphically below:

6.2.1 New pattern for a fixed behavior agent



red green

This new motion pattern may be implemented by adding the following content to the agent's word bank:

6.2.2 Coding motion triggered by red green as a set of proplets

rec: red	rec: green	act: strght	act: left	act: left	act: left	act: left
prv:	prv: red	prv: green	prv: strght	prv: left	prv: left	prv: left
nxt: green	nxt: strght	nxt: left	nxt: left	nxt: left	nxt: left	nxt:
prn: q_1	prn: q_2	prn: q_3	prn: q_4	prn: q_5	prn: q_6	prn: q_7

To automatically derive the new pattern by guiding the agent through the motion shown in 6.2.1, the "record player" system of the LA.act grammar must be complemented by a co-designed "recorder" system, called LA.rec (for recognition). For this the agent must be extended from being able to recognize the stimuli to also being able to recognize and store the elements of the guided responses, resulting in a basic writable memory.

The first step is a definition of lexical proplets, such as the following:

6.2.3 Lexical proplets of an extended fixed behavior agent

rec: red	rec: green	rec: blue	act: strght	act: left	act: right
prv:	prv:	prv:	prv:	prv:	prv:
nxt:	nxt:	nxt:	nxt:	nxt:	nxt:
prn:	prn:	prn:	prn:	prn:	prn:

These proplets are lexical because only the core attributes have values, here the attributes rec (for recognition) and act (for action) .

[6] In machine learning, this would be an example of *learning from instruction* (Mitchell 1997).

Once the fixed behavior agent has been extended to recognize the action steps of certain guided responses, the lexical look-up triggered by guiding the agent through the steps of 6.2.1 will result in the following sequence:

6.2.4 Recognition and lexical look-up of motion pattern 6.2.1

$$\begin{bmatrix}\text{rec: red}\\ \text{prv:}\\ \text{nxt:}\\ \text{prn: q}_1\end{bmatrix} \begin{bmatrix}\text{rec: green}\\ \text{prv:}\\ \text{nxt:}\\ \text{prn: q}_2\end{bmatrix} \begin{bmatrix}\text{act: strght}\\ \text{prv:}\\ \text{nxt:}\\ \text{prn: q}_3\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv:}\\ \text{nxt:}\\ \text{prn: q}_4\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv:}\\ \text{nxt:}\\ \text{prn: q}_5\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv:}\\ \text{nxt:}\\ \text{prn: q}_6\end{bmatrix} \begin{bmatrix}\text{act: left}\\ \text{prv:}\\ \text{nxt:}\\ \text{prn: q}_7\end{bmatrix}$$

To complete this sequence into the proplets of 6.2.2, the core values must be cross-copied to the prev and next slots of adjacent proplets. This is done by the following LA.rec grammar, which uses the same variable definition 6.1.5 as LA.act and establishes semantic relations solely by the order of the input:[7]

6.2.5 Rule system of LA.rec for recording guided patterns

$$\textbf{ST}_S =_{def} \{ (\begin{bmatrix}\text{rec: T}_n\end{bmatrix} \{\text{Rule\_0, Rule\_1}\}) \}$$

Rule_0 {Rule_0, Rule_1}

$$\begin{bmatrix}\text{rec: T}_n\\ \text{nxt:}\\ \text{prn: K}_i\end{bmatrix} \begin{bmatrix}\text{rec: T}_{n+1}\\ \text{prv:}\\ \text{prn: K}_{i+1}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{rec: T}_n\\ \text{nxt: T}_{n+1}\\ \text{prn: K}_i\end{bmatrix} \begin{bmatrix}\text{rec: T}_{n+1}\\ \text{prv: T}_n\\ \text{prn: K}_{i+1}\end{bmatrix}$$

Rule_1 {Rule_2}

$$\begin{bmatrix}\text{rec: T}_n\\ \text{nxt:}\\ \text{prn: K}_i\end{bmatrix} \begin{bmatrix}\text{act: M1}\\ \text{prv:}\\ \text{prn: K}_{i+1}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{rec: T}_n\\ \text{nxt: M1}\\ \text{prn: K}_i\end{bmatrix} \begin{bmatrix}\text{act: M1}\\ \text{prv: T}_n\\ \text{prn: K}_{i+1}\end{bmatrix}$$

Rule_2 {Rule_2}

$$\begin{bmatrix}\text{act: M1}\\ \text{nxt:}\\ \text{prn: K}_i\end{bmatrix} \begin{bmatrix}\text{act: M2}\\ \text{prv:}\\ \text{prn: K}_{i+1}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{act: M1}\\ \text{nxt: M2}\\ \text{prn: K}_i\end{bmatrix} \begin{bmatrix}\text{act: M2}\\ \text{prv: M1}\\ \text{prn: K}_{i+1}\end{bmatrix}$$

$$\textbf{ST}_F =_{def} \{(\begin{bmatrix}\text{nxt: }\end{bmatrix} \text{rp}_{\text{Rule\_2}})\}$$

During a derivation, the parser ensures that all instances of K match the same letter value.

LA.rec is completely general. Given the variable definition 6.1.5, it handles all possible guided patterns consisting of the stimuli red, green, and blue and the response steps strght, left, and right. If there is no restriction on the length of the different sequences and the agent's memory space, their number is infinite. Furthermore, LA.rec will also handle any extensions of the variable definition 6.1.5 (requiring concomitant extensions of the agent's hardware for recognition and/or action, but not of the LA.rec grammar).

---

[7] For an extended system for context recognition see 8.2.2.

LA.act and LA.rec both fit the component structure of diagram 4.5.4: both receive input from the I/O component and both are part of the rule component. LA.act can handle an unlimited number of different sequences stored in the word bank. LA.rec can write an unlimited number of guided pattern sequences into a word bank as long as the available memory space permits.

The meanings of red, green, blue, strgt, left, and right (6.1.5), used by LA.act and LA.rec as core and continuation values, are those of English. Procedurally, these values have straightforward implementations in terms of artificial vision (measuring electromagnetic frequencies) and locomotion. English language intuitions are required only to ensure that the procedural definitions are attached to the correct English surfaces (here used as place holders[8]).

The agent-internal representation of the external world by means of these meanings is extremely sparse – in concord with the tenets of subsumption architecture in robotics (Brooks 1991). For example, a fixed behavior robot trying to execute the sequence 6.2.2 in rough terrain could not possibly model this terrain, given the limits of its recognition. Instead the realization of the motion steps is left to a loosely coupled, massively parallel, analog walking machine as described by Tilden and Hasslacher (1994). This machine is *subsymbolic* because it uses "digital pulse trains ... for motor drive and control" (op. cit.).

For higher-level reasoning, such subsymbolic procedures must be related to symbolic ones. Thereby, nouvelle AI proceeds from the subsymbolic to the symbolic, while DBS proceeds from the symbolic to the subsymbolic. For example, the fixed behavior agent outlined above is symbolic, but assumes a subsymbolic, procedural realization of the core values in terms of the agent's elementary recognition and action procedures.

For learning, however, the relevant transition is not from the symbolic to the subsymbolic or vice versa, but from fixed behavior to adaptive behavior. In contradistinction to expanding the repertoire of a fixed behavior agent by means of guided patterns (provided by a scientist), adaptation and learning must be autonomous, driven by automatic appraisal and pattern derivation.

6.3 Transition from Fixed to Adaptive Behavior

Upscaling a software system of fixed behavior into one of adaptive behavior requires a number of rather obvious extensions:

[8] In this particular respect, the difference between the natural languages boils down to the use of different handles (placeholders) in the form of language-dependent surfaces.

6.3.1 Extensions required by an adaptive behavior agent

1. Writable memory
   In order to record individual recognition action episodes, the agent's non-writable memory must be complemented with a writable memory.
2. Decoupling of recognition and action
   The agent must be capable of *recognition per se*, i.e. recognition without having to perform an associated fixed behavior action, just as there must be action triggered by reasoning rather than by a fixed behavior stimulus.
3. Unknowns
   The agent must be able to recognize and store unknowns consisting of previously unencountered constellations of available recognition elements.[9]
4. Appraisal
   In order to learn from past experiences, the agent must be able to evaluate the implication of recognitions and the outcome of actions.[10]
5. Automatic pattern derivation
   In order to generalize over similar constellations, the agent must be capable of automatic pattern derivation (Sect. 6.4).

For fixed behavior, a non-writable memory is sufficient: once a stimulus-response pattern has been coded, no additional memory is needed for repeating it. Adaptive behavior, however, requires an additional writable memory for recording individual episodes of recognition and action in the order of their arrival.[11] Technically, adaptive and fixed behavior may be combined in the same word bank by using the column of owner values simultaneously for the writable memory to the left and the non-writable memory to the right.

Consider, for example, the fixed behavior agent described in Sect. 6.1. Its word bank has token lines with the owner values blue, green, left, red, right, and strght in alphabetical order. By storing fixed behavior proplets to the right of the owner values and "episodic" proplets to the left, the writable and the non-writable memory may be accessed for storage and retrieval using the same owner values:

[9] An example from vision is *recognition by components* (RBC) based on geons, proposed by Biederman (1987). Sect. 8.6; L&I'05.

[10] Ekman 2003, p. 31, speaks of the automatic appraisal mechanism in the "emotion database" of a cognitive agent. In DBS, appraisal (or evaluation) is integrated into the proplets in the form of values for an additional appraisal attribute; therefore a separate emotion database is not needed.

[11] In nature, practically all living beings, starting with the protozoa, are capable of some form of associative learning (classical conditioning), and must therefore have some writable memory. The guided pattern method for extending a fixed behavior repertoire (Sect. 6.2) also requires sufficient amounts of writable memory, though not for distinguishing individual performances of recognition action sequences.

6.3.2 Writable and non-writable memory in a word bank

$$\begin{array}{llllll}
\multicolumn{3}{c}{\textit{writable memory}} & & & \textit{non-writable memory}\\
 & & & \textit{now front} & \textit{owner values} & \\
\begin{bmatrix}\text{rec: blue}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: 21}\end{bmatrix} & \begin{bmatrix}\text{rec: blue}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: 37}\end{bmatrix} & \begin{bmatrix}\text{rec: blue}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: 55}\end{bmatrix} \ldots & & \text{blue} & \begin{bmatrix}\text{rec: blue}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: } z_1\end{bmatrix}\\
\begin{bmatrix}\text{rec: green}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: 14}\end{bmatrix} & \begin{bmatrix}\text{rec: green}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: 38}\end{bmatrix} & \begin{bmatrix}\text{rec: green}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: 42}\end{bmatrix} \ldots & & \text{green} & \begin{bmatrix}\text{rec: green}\\ \text{prv:}\\ \text{nxt: strght}\\ \text{prn: } y_1\end{bmatrix}
\end{array}$$

For reasons of space, only the first two token lines of the word bank in question are shown. A complete word bank would show all 21 proplets of 6.1.2, 6.1.3, and 6.1.4, sorted into the 6 token lines of the non-writable memory to the right of the owner values.

The writable memory to the left of the owner values is filled by copying[12] each proplet of a sequence performance to the now front. In such episodic proplets, incremented numerical values, e.g. 55, replace the alphabetic prn values, e.g. z_1, of the non-writable memory. The episodic proplets in the writable memory of each token line are in the temporal order of their arrival, as reflected by the increasing prn values, e.g. 21, 37, 55, ...

The extension illustrated in 6.3.2 keeps track of how often a fixed behavior pattern was activated in the past and in what order the activations occurred, but there are no recognitions per se. In order to allow the agent to observe and remember without having to execute an associated fixed behavior, there must also be the possibility of decoupling recognition and action (2 in 6.3.1).[13]

Decoupled elementary items may be derived automatically from any given fixed behavior. Consider, for example, an artificial cognitive agent which continues feeding when it sees a red square, but tries to hide when it recognizes a green circle.

6.3.3 Two examples of alternative fixed behavior patterns

$$1.\quad \begin{bmatrix}\text{rec: red}\\ \text{prv:}\\ \text{nxt: square}\\ \text{prn: } x_1\end{bmatrix} \begin{bmatrix}\text{rec: square}\\ \text{prv: red}\\ \text{nxt: feed}\\ \text{prn: } x_1\end{bmatrix} \begin{bmatrix}\text{act: feed}\\ \text{prv: square}\\ \text{nxt:}\\ \text{prn: } x_1\end{bmatrix}$$

$$2.\quad \begin{bmatrix}\text{rec: green}\\ \text{prv:}\\ \text{nxt: circle}\\ \text{prn: } y_1\end{bmatrix} \begin{bmatrix}\text{rec: circle}\\ \text{prv: green}\\ \text{nxt: hide}\\ \text{prn: } y_1\end{bmatrix} \begin{bmatrix}\text{act: hide}\\ \text{prv: circle}\\ \text{nxt:}\\ \text{prn: } y_1\end{bmatrix}$$

[12] This description is conceptual and does not prescribe technical details of the actual implementation.
[13] Instances of non-decoupled behavior are the reflexes.

Automatic decoupling consists of disassembling the stimulus-response pairs of the agent's fixed behavior into their elementary parts. For example, the disassembling of 6.3.3 results in the following elements, each a separate and manageable task of software and hardware engineering:

6.3.4 Decoupled external recognitions and actions

rec: red	act: hide
rec: green	act: feed
rec: square	
rec: circle	

Any execution of the fixed behavior inferences 6.3.3 is copied into the agent's writable memory in the form of decoupled recognitions and actions as defined in 6.3.4. Thereby, they receive incremented numerical prn values, resulting in their reconnection in terms of their temporal sequence (coordination).

Let us assume that the combinations red square and green circle are knowns, while the alternative combinations green square and red circle are unknowns (3 in 6.3.1). When faced with such unknowns, two kinds of actions are available to the agent, namely act: feed and act: hide. Without additional assumptions, the choice between these options is at random.

For appraisal (4 in 6.3.1), let us add two more decoupled recognitions to 6.3.4, namely internal[14] [rec: good] and internal [rec: bad]. When the agent is faced with an unknown, e.g. red circle, there are the following possibilities:

6.3.5 Possible constellations when faced with an unknown

1	rec: red circle	act: hide	rec: good
2	rec: red circle	act: hide	rec: bad
3	rec: red circle	act: feed	rec: good
4	rec: red circle	act: feed	rec: bad

For example, if the agent recognizes a red circle for the first time, chooses to hide (trial and error, L&I'05), and evaluates the decision as bad (because it interrupts feeding unnecessarily), as in 2, the next[15] encounter of a red circle will not be at random, based on the following general inference rule:

6.3.6 Consequence inference for negative experience (CIN)

rec: α act: β rec: bad **csq** rec: α act: no β

[14] Evaluations are treated as internal recognition.

[15] In real life, agents usually do not change their behavior after the first negative experience. This makes functional sense insofar as different instances of the same behavior pattern may call forth different consequences from the external environment. However, the worse the consequence of an initial random behavior, the more likely the avoidance of that behavior in a second encounter.

Like all DBS inference rules, CIN derives new content (consequent) from given content (antecedent), here using the connective **csq** (for consequence). Unlike the inferences 5.2.4, 5.2.5, 5.3.3, and 5.3.5, the consequent of 6.3.5 is not connected to the antecedent by one or more address values. This is because the consequent may apply to an instantiation of α different from that in the antecedent – though identity at the content level is not precluded.

In a second encounter with a red circle, the antecedent of inference 6.3.6 matches the negative experience 2 in 6.3.5 with a previously unknown. Based on the alternative blueprint for action provided by the consequent of the CIN inference, this second encounter will not result in hiding, thus preventing an unnecessary interruption of the agent's feeding activity.

Applying CIN to the negative experience 4 of 6.3.5 has the opposite effect, causing the agent to hide on next encountering a red circle. The positive experiences 1 and 3 of 6.3.5, in contrast, reenforce the action initially chosen at random, based on the following inference:

6.3.7 CONSEQUENCE INFERENCE FOR POSITIVE EXPERIENCE (CIP)

rec: α act: β rec: good **csq** rec: α act: β

In this way, adaptive behavior derives new action patterns in the agent's writable memory which serve to maintain the agent's balance.[16]

6.4 Upscaling from Coordination to Functor-Argument

The combination of old elementary "knowns" into a new complex "unknown" (3 in 6.3.1), such as red circle in 6.3.5, goes beyond the method of using rec and act as the only core attributes of proplets, practiced for simplicity in the definitions of LA.act (6.1.6) and LA.rec (6.2.5). For example, given that red circle is a functor-argument, with the adjective red modifying the noun circle, we would like to refine the LA.act and LA.rec grammars to handle functor-argument in addition to coordination.

This refinement is parallel to the transition from propositional calculus to first-order predicate calculus in symbolic logic. Symbolic logic constructs the transition by building formulas of predicate calculus from formulas of propositional calculus, using the additional constructs of functors, arguments, variables, and quantifiers. Consider the following example:

6.4.1 ELEMENTS OF PROPOSITIONAL CALC. IN PREDICATE CALCULUS

$[p \wedge q] \implies \exists x[\text{red}(x) \wedge \text{circle}(x)]$

Predicate calculus inherits the connective ∧ (including its truth table) and the bracketing structure which surrounds it from propositional calculus. The unanalyzed propositions p and q of propositional calculus are replaced with the functor-arguments red(x) and circle(x). Like p and q, red(x) and circle(x) denote truth values, even though red circle is linguistically a phrasal noun and not a sentence. The instances of the variable x are horizontally bound by the quantifier ∃x of predicate calculus.

DBS, in contrast, combines coordination with functor-argument within the structural means of the proplet format, as shown below:

6.4.2 Integrating functor-argument in DBS

$$\begin{bmatrix}\text{rec: red}\\ \text{nxt: circle}\\ \text{prv:}\\ \text{prn: 62}\end{bmatrix} \begin{bmatrix}\text{rec: circle}\\ \text{nxt:}\\ \text{prv: red}\\ \text{prn: 62}\end{bmatrix} \Longrightarrow \begin{bmatrix}\text{adj: red}\\ \text{cat: adn}\\ \text{mdd: circle}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 62}\end{bmatrix} \begin{bmatrix}\text{noun: circle}\\ \text{cat: sn}\\ \text{fnc:}\\ \text{mdr: red}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 62}\end{bmatrix}$$

The two proplets preceding the arrow treat red and circle as elementary recognitions,[17] indicated by the core attribute rec. They are coordinated by the values of the continuation attributes next and prev.

In the two proplets following the arrow, the two core attributes rec are revised into the more differentiated adj and noun; the distinction between recognition and action is treated as a property of the core values. The continuation attributes prev and next reappear as nc (for next conjunct) and pc (for previous conjunct); they are available for the intrapropositional coordination of adjectives, as in beautiful, young, intelligent (NLC 8.6.6, 8.6.7), and of nouns, as in the man, the woman, and the child (NLC Sects. 8.2, 8.3).

The determiner properties of the quantifiers ∃x (for some) and ∀x (for all) of predicate calculus are recoded in DBS as values of the sem attribute, i.e. [sem: pl sel] for some and [sem: pl exh] for all. The values pl, sel, and exh stand for plural, selective, and exhaustive, respectively. For a set-theoretical characterization of these values as well as the definition of other determiners like a(n), every, and the see NLC 6.4.7.

The common prn value of the *red* and *circle* proplets, here 62, indicates their belonging to the same proposition, and assumes the binding function of the quantifier in 6.4.1. In the proplets preceding the arrow in 6.4.2, the semantic relation is coordination, expressed by the continuation attributes prev

[16] In machine learning, CIN and CIP would be examples of *learning from experience*.

[17] We refrain from calling an elementary recognition or action a "proposition." A set of proplets with a common prn value is called a proposition only if it contains a verb.

and next. In the proplets following the arrow, the semantic relation is functor-argument in the form of modifier|modified, expressed by the continuation attributes mdr and mdd. The phrasal noun content *red circle* may be turned into a proposition by adding a suitable verb, e.g. *disappear*, connected via the noun's fnc and the verb's arg value.

In addition to the upscaling from coordination only to coordination plus functor-argument there is the upscaling from nonlanguage content only to nonlanguage content plus language content. A language content is created by supplying the nonlanguage proplets' sur attributes with appropriate language-dependent values. Consider the following word bank example containing the content *red circles* as coreferent language and nonlanguage proplets:

6.4.3 Referring with language proplets to context proplets

member proplets				*now front*	*owner values*
. . .					. . .
. . .	sur: noun: circle cat: pnp sem: pl sel fnc: mdr: red nc: pc: prn: 37	. . .	sur: circles noun: (circle 37) cat: pnp sem: pl sel fnc: mdr: (red 37) nc: pc: prn: 62		circle
. . .					
. . .	sur: adj: red cat: adn sem: pad mdd: circle nc: pc: prn: 37	. . .	sur: red adj: (red 37) cat: adn sem: pad mdd: (circle 37) nc: pc: prn: 62		red

Compared to treating reference as a "vertical matching" between language and context proplets (4.3.3[18]), the above example takes a database view by sorting proplets into token lines solely according to their core value, regardless of whether there is a sur value (language proplet) or not (context proplet).

The result is a treatment of reference as a "horizontal" relation between language and context proplets within the same token line, in accordance with the refined component structure of a cognitive agent presented in 4.5.3 and 4.5.4. The distinction between language and context proplets is still available in 6.4.3, however. The language proplets with the prn value 62 have the

[18] Examples 4.3.3 and 6.4.3 differ not only in their core values, but also in that the former shows reference based on similarity, using the type/token relation, while the latter shows coreference, using addresses for coding identity.

nonempty sur values red and circles, whereas the corresponding sur slots in the context proplets with the prn value 37 have no value, represented by empty space. The order of content and language proplets in a token line is determined solely by the order of their arrival.

In summary, a language proplet which is used literally occurs in the same token line as its referent. A coreferent item always follows the original. Reference may be based on similarity or identity. Similarity is coded by using a concept type in the coreferent item and a concept token in the referent (4.3.3), while identity is coded by using the address of the referent as the core value of the coreferent item (6.4.3).

The uniform coding of language and nonlanguage proplets is in concord with the Humboldt-Sapir-Whorf hypothesis, according to which a natural language influences the thought of its speaker-hearers. DBS adopts also the reverse direction by treating natural language expressions as a direct reflection of thought. This has the following impact on the computational treatment of reference in DBS (Sect. 1.6):

6.4.4 Fifth Mechanism of Communication (MoC-5)

> Exporting coordination and functor-argument from language content to nonlanguage (context) content, coded uniformly as sets of proplets, facilitates implementing *reference* as a pattern matching between language and context content in the hear and the speak mode.

The close relation between the language and the context level constitutes a form of linguistic relativism (Sect. 3.6). It is moderated by the hypothesis that all natural languages function the same in communication (DBS universals 1.2.1–1.2.4) and is realized concretely by using the same general DBS software machine for different languages (3.5.2).[19]

6.5 Pattern Derivation and Hierarchy Inferencing

By replacing (i) the variable-binding function of the logical quantifiers with a shared prn value and (ii) the determiner function of $\exists x$ and $\forall x$ with sem values (NLC 6.4.7), DBS limits the use of variables to pattern proplets. Content proplets, in contrast may by definition not contain any variables. The purpose of pattern proplets is the matching of corresponding content proplets (Sects. 3.2, 5.4).

[19] See also NLC Sect. 4.6, for a distinction between universal and language-dependent properties.

A pattern may be automatically derived from a content by replacing constants with variables (simultaneous substitution). The set of content proplets matched by a pattern is called its *yield*. The yield of a pattern relative to a given word bank may be controlled precisely by two complementary methods. One is by the choice and number of constants in a content which are replaced by variables. The other is by variable restrictions.

The use of restricted variables allows us to convert any content into strictly equivalent patterns and any pattern into strictly equivalent contents. As an example of the content-pattern conversion consider the content corresponding to Every[20] child slept. Fido snored., i.e. the coordination of two propositions, each with a subject/predicate construction, one with a phrasal noun, the other with a proper name as subject.

6.5.1 CONVERTING A CONTENT INTO AN EQUIVALENT PATTERN

$$\textit{pattern}\quad \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: snp}\\ \text{sem: pl exh}\\ \text{fnc: }\beta\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past}\\ \text{arg: }\alpha\\ \text{mdr:}\\ \text{nc: (}\delta\text{ K+1)}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: }\gamma\\ \text{cat: snp}\\ \text{sem: nm sg}\\ \text{fnc: }\delta\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K+1}\end{bmatrix} \begin{bmatrix}\text{verb: }\delta\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past}\\ \text{arg: }\gamma\\ \text{mdr:}\\ \text{nc:}\\ \text{pc: (}\beta\text{ K)}\\ \text{prn: K+1}\end{bmatrix} \quad \begin{array}{l}\text{where } \alpha \in \{\text{child}\},\ \beta \in \{\text{sleep}\},\\ \gamma \in \{\text{Fido}\},\ \delta \in \{\text{snore}\}, \text{ and}\\ \text{K} \in \{26\}\end{array}$$

$$\Longleftrightarrow$$

$$\textit{content}\quad \begin{bmatrix}\text{noun: child}\\ \text{cat: snp}\\ \text{sem: pl exh}\\ \text{fnc: sleep}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 26}\end{bmatrix} \begin{bmatrix}\text{verb: sleep}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past}\\ \text{arg: child}\\ \text{mdr:}\\ \text{nc: (snore 27)}\\ \text{pc:}\\ \text{prn: 26}\end{bmatrix} \begin{bmatrix}\text{noun: Fido}\\ \text{cat: snp}\\ \text{sem: nm sg}\\ \text{fnc: snore}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 27}\end{bmatrix} \begin{bmatrix}\text{verb: snore}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past}\\ \text{arg: Fido}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc: (sleep 26)}\\ \text{prn: 27}\end{bmatrix}$$

In this example, all core and prn values of the content are simultaneously substituted with variables in the pattern (method one) and all variables are restricted to the value they replace (method two). In this way, strict equivalence between the content and the pattern representation is obtained.

The yield of a pattern may be increased by adding values to the restriction sets of variables. Consider the following example of a pattern matching the content corresponding to Every child slept, but with extended variable restrictions.

---

[20] The fact that every takes a singular noun but refers to a plural set is expressed by the cat value snp, for singular noun phrase, and the sem values pl exh. Phrasal noun and name proplets are shown with the same attribute structure, but different cat and sem values. Compare the special value structure characteristic of proper name proplets described in HBTR Sect. 3.4 with Montague's type raising in lambda calculus 1974, PTQ.

6.5.2 Converting a pattern into equivalent contents

$$\textit{pattern}\quad \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: snp}\\ \text{sem: pl exh}\\ \text{fnc: }\beta\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past}\\ \text{arg: }\alpha\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\quad \begin{array}{l}\text{where } \alpha \in \{\text{man, woman, child}\}, \beta \in \{\text{sleep, sing, dream}\}\\ \text{and K} \in \mathbb{N}\end{array}$$

$\Longleftrightarrow$

	Every man slept.	Every man sang.	Every man dreamed.
content	Every woman slept.	Every woman sang.	Every woman dreamed.
	Every child slept.	Every child sang.	Every child dreamed.

The contents are generated from the pattern by systematically replacing the variables α and β with elements of their restriction sets. Even if no restriction set is specified, the yield of a pattern relative to an actual word bank will still be finite. The ϵ operator connecting a variable with its restriction set is used here in the interpretation "may be instantiated as."

A pattern may be derived automatically from any set of partially overlapping contents. Consider the following example:

6.5.3 Set of contents with partial overlap

Julia eats an apple	John eats an apple	Suzy eats an apple	Bill eats an apple
Julia eats a pear	John eats a pear	Suzy eats a pear	Bill eats a pear
Julia eats a salad	John eats a salad	Suzy eats a salad	Bill eats a salad
Julia eats a steak	John eats a steak	Suzy eats a steak	Bill eats a steak

Of these 16 propositions, each contains the proplet *eat*, while the proplets *Julia, John, Suzy*, and *Bill* occur four times as subject and the proplets *apple, pear, salad*, and *steak* occur four times as object. Based on these repetitions, the propositions may be summarized as the following pattern:

6.5.4 Summarizing the set 6.5.3 as a pattern

$$\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: eat}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: eat}\\ \text{arg: }\alpha\ \beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: }\beta\\ \text{fnc: eat}\\ \text{prn: K}\end{bmatrix}$$

where $\alpha \in$ {Julia, John, Suzy, Bill} and $\beta \in$ {apple, pear, salad, steak}

Due to the restriction on the variables α and β, 6.5.4 is strictly equivalent to 6.5.3. From a linguistic point of view, 6.5.4 may be seen as the valency pattern (Ágel et al. 2006) or lexical frame of the transitive verb *eat*. The restriction sets of the variables α and β may be established automatically by parsing a corpus (Sect. 15.5): all subjects of *eat* actually occurring in the corpus are written into the restriction set of α and all objects are written into the restriction set of β.

One of the many results of automatic pattern derivation is the *is-a* hierarchy (subsumptive containment hierarchy) familiar from knowledge representation and object-oriented programming.[21] In DBS, the subclass relation between food as the hypernym and the set apple, pear, salad, steak as the instantiation, for example, may be coded by extending the formal restriction technique for variables (6.5.1, 6.5.4) to certain constants:

6.5.5 Coding the subclass relation for food

$$\begin{bmatrix}\text{noun: food}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix}$$

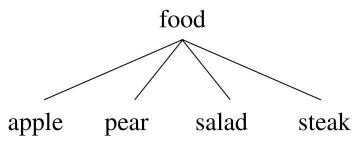
where *food* ϵ {apple, pear, salad, steak}

As in the restriction of variables, ϵ is used here in the sense "is instantiated as." The hypernym concept *food* may serve as the content (literal meaning$_1$) attached to the word form surface food in English, aliment in French, Nahrung in German, cibo in Italian, and so on (2.1.1, 3.1.1).

The derivation of a semantic hierarchy is empirically adequate if the resulting class containing the instantiations corresponds to that of the surrounding humans. For example, if the artificial agent observes humans to habitually (frequency) eat müsli, the restriction list of *food* must be adjusted correspondingly.[22] Furthermore, the language surface, e.g. aliment, chosen by the artificial agent for the hypernym concept, e.g. *food*, must correspond to that of the natural language in use.

Implicit in pattern 6.5.5 is the following tree structure, which a substitution approach might formalize as the rewrite rule food $\rightarrow$ apple, pear, salad, steak.

6.5.6 Representing the semantic hierarchy 6.5.5 as a tree



Just as such a tree requires some tree-walking algorithm to get from a higher node to a lower node or vice versa, a pattern like 6.5.5 requires DBS inferences to utilize the content of the hierarchy for reasoning. For example, an inference for downward traversal should allow the agent to infer that *food* may

[21] As a subclass relation, the *is-a* hierarchy is a prime example of the substitution (as opposed to the continuation) approach because it is motivated more directly and more obviously than the constituent structure trees (Sect. 12.2) of phrase structure grammar and categorial grammar.

be instantiated by apple, pear, salad, or steak, while an inference for upward traversal should infer that an apple, for example, instantiates *food*.[23]

Given that semantic hierarchy relations abound in the lexicon, we derive the inferences for their upward and downward traversal automatically by means of a meta-inference which takes a pattern like 6.5.5 as input and derives the associated inferences for upward and downward traversal:

6.5.7 Meta-inference deriving **down** and **up** inferences

$$\begin{array}{lcll}
\textit{antecedent} & & \textit{consequent 1} & \textit{consequent 2} \\
\begin{bmatrix}\text{noun: HT}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} & \Longrightarrow & \begin{bmatrix}\text{noun: HT}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \textbf{ down } \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }(\beta\ \text{K})\\ \text{prn: K+M}\end{bmatrix} & \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \textbf{ up } \begin{bmatrix}\text{noun: HT}\\ \text{fnc: }(\beta\ \text{K})\\ \text{prn: K+M}\end{bmatrix} \\
\text{where HT } \epsilon \text{ \{A, B, C, D, ...\}} & & \text{where } \alpha\ \epsilon \text{ \{A, B, C, D, ...\}} & \text{where } \alpha\ \epsilon \text{ \{A, B, C, D, ...\}}
\end{array}$$

The variable HT stands for a higher-term constant, e.g. *food*, while the variables A, B, C, D, . . . , stand for the instantiation constants, e.g. apple, pear, salad, and steak. Other examples of higher term constants are *mammal*, which is instantiated by dogs, cats, and mice; *vehicle*, which is instantiated by cars, trucks, and buses; *water*, which is instantiated by ocean, lake, and river; and so on. They may all be derived automatically in the same way as is the higher term *food* in 6.5.3–6.5.5.

Application of the meta-inference 6.5.7 to the *food* pattern 6.5.5 results in the following inference for downward traversal:

6.5.8 Applying meta-inference 6.5.7 to derive **down** inference

$$\begin{array}{lcl}
\textit{antecedent} & & \textit{consequent} \\
\begin{bmatrix}\text{noun: HT}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} & \Longrightarrow & \begin{bmatrix}\text{noun: HT}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \textbf{ down } \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }(\beta\ \text{K})\\ \text{prn: K+M}\end{bmatrix} \\
\text{where HT } \epsilon \text{ \{A, B, C, D, ...\}} & & \text{where } \alpha\ \epsilon \text{ \{A, B, C, D,...\}} \\
\Uparrow \textit{ matching and binding} & & \Downarrow \\
\begin{bmatrix}\text{noun: food}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} & & \begin{bmatrix}\text{noun: food}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \textbf{ down } \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }(\beta\ \text{K})\\ \text{prn: K+M}\end{bmatrix} \\
\text{where food } \epsilon \text{ \{apple, pear, salad, steak\}} & & \text{where } \alpha\ \epsilon \text{ \{apple, pear, salad, steak\}}
\end{array}$$

While meta-inferences are applied to pattern proplets with restricted higher terms as core values, the resulting inferences are applied to content proplets, i.e. proplets without any variables.

For example, if the content Julia is looking for food is activated, the inference for downward traversal 6.5.8 may be applied to it as follows:

---

[22] This resembles the establishment of inductive inferences in logic, though based on individual agents.
[23] Hypernyms in different agents may vary, depending on personal preferences in different cultures.

6.5.9 Applying inference for downward traversal

$$\begin{array}{lllll}
 & & & \textit{antecedent} & & \textit{consequent} \\
\textit{pattern level:} & & & \begin{bmatrix}\text{noun: food}\\ \text{fnc: } \beta \\ \text{prn: K}\end{bmatrix} & \textbf{down} & \begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } (\beta \text{ K}) \\ \text{prn: K+M}\end{bmatrix} \\
 & & & \multicolumn{3}{l}{\text{where } \alpha \in \{\text{apple, pear, salad, steak}\}} \\
 & & \multicolumn{2}{r}{\textit{matching and binding} \Uparrow} & & \Downarrow \\
\textit{content level:} & \begin{bmatrix}\text{noun: Julia}\\ \text{fnc: look_for}\\ \text{prn: 18}\end{bmatrix} & \begin{bmatrix}\text{verb: look_for}\\ \text{arg: Julia food}\\ \text{prn: 18}\end{bmatrix} & \begin{bmatrix}\text{noun: food}\\ \text{fnc: look_for}\\ \text{prn: 18}\end{bmatrix} & & \begin{bmatrix}\text{noun: } \alpha \\ \text{verb: (look_for 18)}\\ \text{prn: 25}\end{bmatrix}
\end{array}$$

When the antecedent, consisting of a single pattern proplet with the core value food, matches a corresponding proplet at the content level, the consequent derives a new content containing the following disjunction[24] of several proplets with core values corresponding to the elements of the restriction set of α:

6.5.10 Output disjunction of the downward inference 6.5.8

$$\begin{bmatrix}\text{noun: apple\&}\\ \text{fnc: (look\_for 18)}\\ \text{nc: pear}\\ \text{prn: 25}\end{bmatrix} \begin{bmatrix}\text{noun: pear}\\ \text{pc: apple\&}\\ \text{nc: salad}\\ \text{prn: 25}\end{bmatrix} \begin{bmatrix}\text{noun: salad}\\ \text{pc: pear}\\ \text{nc: steak}\\ \text{prn: 25}\end{bmatrix} \begin{bmatrix}\text{noun: steak}\\ \text{sem: } \textit{or}\\ \text{pc: salad}\\ \text{prn: 25}\end{bmatrix}$$

The proplets of the output disjunction are concatenated by the pc (previous conjunct) and nc (next conjunct) features, and have the new prn value 25. They are related to the original proposition by the address (look_for 18) serving as the fnc value of the first disjunct. The output disjunction may be completed automatically into the new proposition Julia looks_for apple or pear or salad or steak, represented as follows:

6.5.11 Proposition resulting from downward inference 6.5.9

$$\begin{bmatrix}\text{noun: (Julia 18)}\\ \text{fnc: (look\_for 18)}\\ \text{prn: 25}\end{bmatrix} \begin{bmatrix}\text{verb: (look\_for 18)}\\ \text{arg: (Julia 18) apple\&}\\ \text{prn: 25}\end{bmatrix} \begin{bmatrix}\text{noun: apple\&}\\ \text{fnc: (look\_for 18)}\\ \text{nc: pear}\\ \text{pc:}\\ \text{prn: 25}\end{bmatrix} \begin{bmatrix}\text{noun: pear}\\ \text{fnc:}\\ \text{pc: apple\&}\\ \text{nc: salad}\\ \text{prn: 25}\end{bmatrix} \begin{bmatrix}\text{noun: salad}\\ \text{fnc:}\\ \text{pc: pear}\\ \text{nc: steak}\\ \text{prn: 25}\end{bmatrix} \begin{bmatrix}\text{noun: steak}\\ \text{sem: } \textit{or}\\ \text{fnc:}\\ \text{pc: salad}\\ \text{nc:}\\ \text{prn: 25}\end{bmatrix}$$

This new proposition with the prn value 25 is derived from the input proposition with the prn value 18 shown at the content level of 6.5.9, and related to it by pointer values.

The inverse of downward traversal is the upward traversal of a semantic hierarchy. An upward inference assigns a hypernym like *food* to concepts like *salad* or *steak*. Consider the following application of the inference for upward traversal 6.5.7 (*consequent 2*).

---

[24] See NLC Chap. 8, for a detailed discussion of intrapropositional coordination such as conjunction and disjunction.

6.5.12 Hierarchy-inference for upward traversal

$$
\begin{array}{lccccc}
 & \textit{antecedent} & & & & \textit{consequent} \\
\textit{pattern level:} & \alpha \in \{\text{apple, pear, salad, steak}\} & \& & \begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } \beta \\ \text{prn: K}\end{bmatrix} & \textbf{up} & \begin{bmatrix}\text{noun: food} \\ \text{fnc: } (\beta \text{ K}) \\ \text{prn: K+M}\end{bmatrix} \\
 & & & \textit{matching and binding} & & \\
\textit{content level:} & \begin{bmatrix}\text{noun: Julia} \\ \text{fnc: prepare} \\ \text{prn: 23}\end{bmatrix} \begin{bmatrix}\text{verb: prepare} \\ \text{arg: Julia salad} \\ \text{prn: 23}\end{bmatrix} & & \begin{bmatrix}\text{noun: salad} \\ \text{fnc: prepare} \\ \text{prn: 23}\end{bmatrix} & & \begin{bmatrix}\text{noun: food} \\ \text{fnc: (prepare 23)} \\ \text{prn: 29}\end{bmatrix}
\end{array}
$$

As in the downward inference 6.5.9, the antecedent of the upward inference consists of a single pattern proplet, now with the restricted variable α as the core value. Due to the use of an address as the fnc value of the output, there is sufficient information to complete the output proplet into the new proposition Julia prepares food (not shown), with the prn value 29 and the extrapropositional address values (Julia 23) and (prepare 23).

The automatic derivation and restriction of patterns like 6.5.4 and 6.5.5 directly controls the automatic adaptation of the hierarchy inferences by adjusting their restriction sets. In this way, DBS fulfills the three functions which define an autonomic system: "automatically configure itself in an environment, optimize its performance using the environment and mechanisms for performance, and continually adapt to improve performance and heal itself in a changing environment" (Naphade and Smith 2009).

6.6 Natural vs. Artificial Language Learning

The similarity between natural and artificial cognitive agents regarding their theoretical (declarative) structure and the dissimilarity in their practical (procedural) implementation, i.e. hardware vs. wetware, may be illustrated with language acquisition. Take for example the learning of word forms.

For humans, this is a slow process, taking several years in childhood (first language acquisition) and in adulthood (second language acquisition). An artificial agent, in contrast, may simply be uploaded with an online dictionary and the associated software for automatic word form recognition – not just for one language, but for as many languages as available or desired.

The word form analyses provided in this way to the artificial agent specify (i) the morphosyntactic properties formalized as proplet shells, (ii) the language-dependent surface, and (iii) the core values as placeholders using their English counterparts. The proplet shells and the core values are orthogonal to each other in the sense that (a) a given proplet shell may take different core values and (b) a given core value may be embedded into different proplet shells.

The following example shows a proplet shell taking different core values:

6.6.1 Proplet shell taking different core values

$$
\begin{array}{cccccc}
\textit{proplet shell} & \textit{context proplets} & & & & \\
\begin{bmatrix} \text{sur:} \\ \text{noun: } \alpha \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix}
\Rightarrow &
\begin{bmatrix} \text{sur:} \\ \text{noun: dog} \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix} &
\begin{bmatrix} \text{sur:} \\ \text{noun: book} \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix} &
\begin{bmatrix} \text{sur:} \\ \text{noun: child} \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix} &
\begin{bmatrix} \text{sur:} \\ \text{noun: apple} \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix} & <
\end{array}
$$

The proplets derived from the proplet shell differ in only one value (see <), which should facilitate learning. The proplets are lexical proplets because of their empty continuation attributes fnc, mdr, nc, and pc as well as their empty book-keeping attribute prn. The proplets are context proplets because their sur attribute has no value.

Context proplets may be turned into language proplets by inserting the appropriate sur values (see <), as in the following example for English:

6.6.2 Turning context proplets 6.6.1 into language proplets

$$
\begin{array}{cccccc}
\textit{proplet shell} & \textit{language proplets} & & & & \\
\begin{bmatrix} \text{sur: } \alpha\text{'+x} \\ \text{noun: } \alpha \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix}
\Rightarrow &
\begin{bmatrix} \text{sur: dog+s} \\ \text{noun: dog} \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix} &
\begin{bmatrix} \text{sur: book+s} \\ \text{noun: book} \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix} &
\begin{bmatrix} \text{sur: child+ren} \\ \text{noun: child} \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix} &
\begin{bmatrix} \text{sur: apple+s} \\ \text{noun: apple} \\ \text{cat: pn} \\ \text{sem: pl} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix} & <
\end{array}
$$

Assuming that the context proplets in 6.6.1 have been acquired already, learning the associated language proplets involves only a single value, namely that of the sur attribute, again facilitating learning.

Once the proplets have been acquired for one language, they may be reused for other languages, provided the lexicalization is similar.[25] The following example shows the proplets for the concept *dog* with English, French, German, Italian, and Polish surfaces:

[25] See 3.6.1 for examples of different lexicalizations. Other examples are (i) German Traumreise (literally *dream journey*), which has been translated into American English as dream vacation and into French as voyage des rêves, (ii) English horseshoe, which translates into German as Hufeisen (literally *hoof iron*) and into French as fer à cheval (literally *iron for horse*), and (iii) French ralenti, which translates into English as slow motion and into German as Zeitlupe.

6.6.3 Taking sur values from different languages

proplet shell *language proplets*

$$\begin{bmatrix}\text{sur: }\alpha' \\ \text{noun: }\alpha \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{sur: dog} \\ \text{noun: dog} \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} \begin{bmatrix}\text{sur: chien} \\ \text{noun: dog} \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} \begin{bmatrix}\text{sur: Hund} \\ \text{noun: dog} \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} \begin{bmatrix}\text{sur: cane} \\ \text{noun: dog} \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} \begin{bmatrix}\text{sur: pies} \\ \text{noun: dog} \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} <$$

For syntactic-semantic parsing, the French, German, Italian, and Polish proplet versions will have to be complemented with the additional cat value m (for the grammatical gender masculine). This language-dependent information may be obtained from the traditional dictionaries for these languages. In addition, corpus-based information, such as domain-dependent frequency, predecessors and successors ordered according to frequency (n-grams), semantic relations, etc., may be added (Sect. 15.5).

In addition to the orthogonal relationship between a given proplet shell and different core values (6.6.1) there is the inverse orthogonal relationship between a given core value and different proplets shells. Embedding a core value into different proplet shells is a simple but effective method to enhance the expressive power of the lexicon of a natural language without having to acquire additional core values. For example, the core value *book* may be used as a noun, a verb, or an adj:

6.6.4 Examples using *book* in different core attributes

Mary loves a good *book* (noun).
Mary *book*ed (verb) a flight to Paris.
Mary is a rather *book*ish (adj) girl.

The lexical *book* proplets used in these contents share the same core value (see <) and are defined as follows:

6.6.5 Core value *book* in noun, verb, and adj proplets

$$\textit{book} \Longrightarrow \begin{bmatrix}\text{sur: book} \\ \text{noun: book} \\ \text{cat: sn} \\ \text{sem: sg} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} \begin{bmatrix}\text{sur: booked} \\ \text{verb: book} \\ \text{cat: n}'\text{ a}'\text{ v} \\ \text{sem: past} \\ \text{arg:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} \begin{bmatrix}\text{sur: bookish} \\ \text{adj: book} \\ \text{cat: adn} \\ \text{sem: pad} \\ \text{mdd:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:}\end{bmatrix} <$$

Similar examples are *red* and *square*, which may also be used as the core values of a noun, a verb, and an adj, as in the following contents:

6.6.6 Examples using *red* and *square* in different core attributes

Mary preferred the other *red* (noun).
The rising sun *red*dened (verb) the sky.
Mary drank *red* (adj) wine.

Mary's house faces a *square* (noun).
Mary *square*d (verb) her account.
Mary bought a *square* (adj) table.

The lexical methods of using (i) different core (6.6.1) and surface (6.6.2) values in the same proplet shell and (ii) the same core value in different proplet shells (6.6.5) are complemented by (iii) syntactic-semantic composition, resulting in an enormous increase in expressive power. For example, embedding the core values *book, square,* and *red* into V(erb), N(oun), and A(djective) proplet shells allows the formal DBS construction of the following contents:

6.6.7 Core values in syntactic-semantic composition

book_V the red_A square_N
book_V the square_N red_A
book_V the square_A red_N
square_V the red_A book_N
square_V the book_N red_A
square_V the book_A red_N
redden_V the square_A book_N
redden_V the book_N square_A
redden_V the book_N square_N ...

The examples are all grammatically well-formed and their core values may be implemented procedurally[26] as the recognition patterns of a talking robot.

The dadaistic absurdity of the literal meanings_1 of the expressions in 6.6.7 highlights the cognitive mechanism of the compositional semantics in natural language. It demonstrates the need to distinguish the literal $\textit{meaning}_1$ of language expressions from the speaker $\textit{meaning}_2$ of utterances. The latter, defined

[26] The definition of basic concepts (core values) as recognition and action procedures provides a *grounding* of the semantics and constitutes a fundamental difference between the agent-based approach of DBS and the sign-based approach of truth-conditional semantics (NLC Sect. 6.4).

in Pop-1 (1.4.1; FoCL 4.3.3) as the use of the former relative to a context of interpretation, seem mostly to fail for 6.6.7. Which of the meanings$_1$ can be used literally or non-literally, or not be used sensibly at all, depends on the actual context of interpretation, for humans and talking robots alike.

Because linguistic examples as isolated signs do not have any concrete context of interpretation, evaluating the utterance meaning$_2$ of the examples 6.6.7 amounts methodologically to evaluating how easily they can be supplied with an imagined (virtual) context of interpretation. Especially for non-literal uses, the result depends on the fantasy of the evaluating agent.

The literal meaning$_1$ of signs, in contrast, (i) are independent of utterances and (ii) must be essentially the same in order for communication between two agents to work. This stability of core values througout the language community allows the use of placeholders as temporary substitutes for their procedural implementation in the following cognitive procedures:

6.6.8 Cognitive procedures using placeholder core values

1. The time-linear syntactic-semantic interpretation in the hear mode (3.3.1),
2. the storage of content provided by recognition and inferencing in the word bank (4.4.1),
3. the navigation-based semantic-syntactic production in the speak mode (Sect. 7.4),
4. the definition of such language dependent lexical relations as synonymy, antonymy, hypernymy, hyponymy, meronymy, and holonymy as well as of such language independent non-lexical relations as cause-effect (Sect. 5.3),
5. the design and implementation of reactor, deductor, and effector inferences (Sect. 5.1),
6. the design and implementation of language inferences for adjusting perspective (Chaps. 10 and 11), and
7. the interaction between the context and language levels (Sect. 4.3).

It follows that computational linguistics can provide a talking robot with a functionally complete software framework of cognition using only placeholder values. In this way the necessary implementation of artificial recognition and action in an autonomous robot (NLC 2.5.2) may be postponed until it may eventually be provided by the colleagues in robotics.

In other words, place holder core values allow an unhindered upscaling of the theoretical software model, but are insufficient for building a talking robot:

without the procedural counterparts, the artificial cognitive agent can neither understand[27] language nor act meaningfully.

Conversely, without a functional software reconstruction of cognition, the robots external interfaces will have no place to map into and out of. Procedurally defined core values are needed (i) for interpreting the input provided by the agent's external recognition interfaces, e.g. eyes and ears, (ii) controlling the output of the agent's external action interfaces, e.g. hands, legs, and vocal tract, (iii) as the basic concepts of the agent's context level content, and (iv) as the agent's basic language meanings.

In practice, this means that learning a new word requires the agent not only to remember the correct surface, select the correct proplet shell, and enter the correct placeholder core value, but also to acquire the core value's correct procedural implementation. Thereby, complex core value procedures may be built from elementary ones and often be acquired by observation as a form of contextual definition.

Consider, for example, an artificial agent taken to the zoo where it sees a herd of zebras for the first time. By looking at a certain zebra several times and comparing it with the other zebras, the agent automatically derives a new type by distilling the necessary properties within certain ranges, leaving aside the accidental properties of the individuals.[28]

Instead of constructing a type holistically from scratch, it may also be constructed compositionally. For example, if the type for horse is already available to the agent, the new type in question may be constructed as a combination of the types for small horse and for black-and-white stripes. If the type for horse is not available, in contrast, the type for zebra may be constructed from more basic types, e.g. for head, body, legs, tail, and black-and-white stripes – such that a later first encounter with a horse will result in a type defined as *big zebra without black-and-white stripes*.

The alternative definitions of a zebra in terms of a horse or a horse in terms of a zebra are equivalent for all practical purposes. Either type will allow the agent to recognize any future zebras, resulting in instantiations (zebra tokens) serving as the procedural core values of context proplets. In this way, an artificial agent without language may learn an open number of new concepts.[29] In vision, these concepts are built up as new combinations of elementary, universal types implemented as line, edge, ridge, color, etc. detectors (Sect. 8.6).

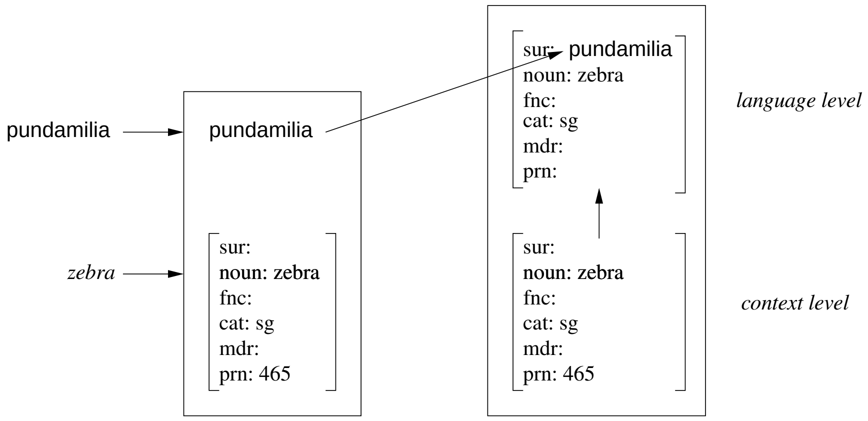
[27] In the sense of visualizing, for example, the difference between *red* and *blue*.

[28] In machine learning, this kind of learning is called learning by observation.

[29] In other words, it is not necessary in DBS to bootstrap all cognition from a small set of universal, basic language concepts, in contradistinction to the proposals by Wierzbicka (1991).

Next consider an artificial agent with language learning a new word. After the agent has acquired the concept type for a zebra, a human points at the zebras and utters the Swahili word pundamilia.[30] Given the similarity between context proplets and language proplets, the artificial agent is able to extend its Swahili vocabulary by (i) copying the context proplet to the language level and (ii) inserting the Swahili surface into the sur slot of the copied proplet.

6.6.9 Learning a new word



In this way, an artificial agent may learn an open number of words by contextual definition – where "contextual" means the (nonlanguage) context of interpretation (and not the language-level co-text).

Seen from the outside, the behavior of an autonomous agent has a substantial grade of unpredictability. In humans, the inherent uncertainty about another's state of mind is buffered by customs of politeness. Politeness must be realized also in the behavior of an artificial cognitive agent. Given that politeness is ritualized (fixed), this should not be too difficult.

Seen from the inside, in contrast, the behavior of an agent is a constant effort to maintain a state of balance. In a DBS system, any parameter values deviating from the "desired" value (Sollwert) are recorded and used as triggers to compute compensatory actions. For the scientist, the fine-tuning of this process is facilitated by the service channel (Sect. 2.4), which provides direct access to the operations of the artificial agent's cognition.

The behavior patterns emerging over time provide for efficient performance as long as environments are relatively stable and are necessary as a process of fitting in. However, the agent must also have the ability to cope with the

[30] The Swahili surface for Zebra was chosen here because it clearly differs from the core value (i.e. the English place holder), similar to the examples in 6.6.3.

unexpected. In DBS, innovative behavior results from the automatic derivation and application of inferences[31] (Sect. 6.5); the continuous interaction between habitual behavior and innovative behavior relies on their uniform coding as sets of proplets connected by address and their time-linear processing.

[31] A cognitively less demanding option is the emulation of others.

Part II

Coding of Content

7. Compositional Semantics

The design of a cognitive agent requires (i) completeness of *function*, (ii) completeness of *data coverage*, and (iii) performance in *real time*. Requirements (i) and (ii) are essential for adapting the agent to its ecological niche; requirement (iii) is essential for success in the daily struggle for survival.

Part I approached functional completeness by modeling the mechanism of natural language communication in the form of a DBS robot with a body, a memory, external interfaces, and a functional flow from input to output. In Part II we turn to completeness of data coverage. The data to be analyzed are kinds of *content*, used in nonlanguage and language cognition alike.

7.1 Forms of Graphical Representation

As an agent-oriented approach, DBS provides separate treatments for the hear and the speak mode: the hear mode maps raw input data, i.e. unanalyzed agent-external language surfaces, into content, while the speak mode maps content into raw output data. A content is represented as a set of proplets concatenated with semantic relations of structure defined by address. The semantic relations of structure are classic functor-argument and coordination.

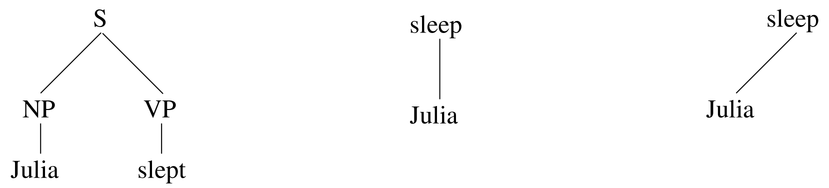
As the first step towards a straightforward translation into efficiently running software, empirical linguistic work on data coverage is aided by simple graphical formats for showing the distinctive properties of a wide variety of grammatical structures. For the hear mode, this format was illustrated in 3.3.1. For the speak mode, a preliminary format was shown in 3.3.3.

We turn now to an advanced format which shows the semantic relations graphically. The following example compares the DBS graph analysis of the subject/predicate[1] relation in Julia slept with the corresponding tree structures of phrase structure grammar (PSG) and dependency grammar (DG):

[1] Terminologically, *predicate* goes with *subject* and *object*, while *verb* goes with *noun*, pace Greenberg (1963). The former terms refer to semantic functions which may be realized by the latter.

7.1.1 DIFFERENT REPRESENTATIONS OF SUBJECT-PREDICATE RELATION

(i) phrase structure grammar *(ii) dependency grammar* *(iii) DBS*



The lines (edges) between the nodes (vertices) have completely different interpretations in the three graphs.

In (i) phrase structure grammar (PSG), the lines specify *dominance* and *precedence*. For example, S dominates NP and VP, and NP precedes VP. Thus, the subject/predicate relation is not expressed explicitly[2] in the PSG graph, but must be deduced from the part of speech interpretation of the nodes NP and VP, and their configuration.

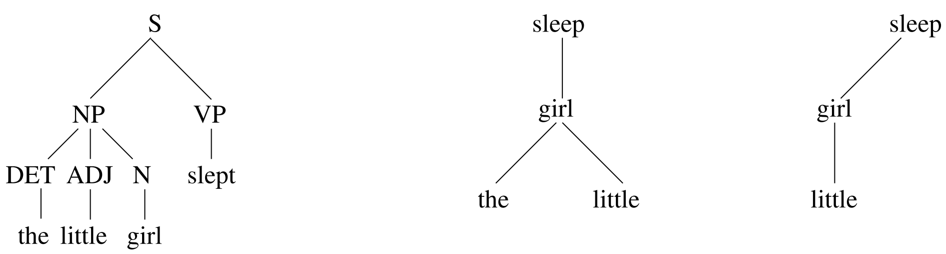
In (ii) dependency grammar (DG), the line between sleep and Julia specifies Julia as *dependent* on sleep. This does not distinguish between the subject/predicate and the object\ predicate relation, which are analyzed equally as a dependency.[3]

In (iii) DBS, the edge "/" is used to indicate the subject/predicate relation between Julia and sleep. Altogether, there are four kinds of lines (edges):[4] in addition to (i) "/" for subject/predicate, there is (ii) "\" for object\predicate, (iii) "|" for modifier|modified, and (iv) "−" for conjunct−conjunct (coordination). Furthermore, there are three core attributes (Sect. 3.5), which are represented by the nodes (vertices) N for noun, V for verb, and A for adjective.

As another comparison, consider the graphical analyses of the similar but somewhat more complex sentence The little girl slept in (i) PSG, (ii) DG, and (iii) DBS. Here, the additional semantic relation of structure is adnominal modification, represented in DBS as the modifier|modified:

7.1.2 COMPARING DETERMINER-ADJECTIVE-NOUN CONSTRUCTIONS

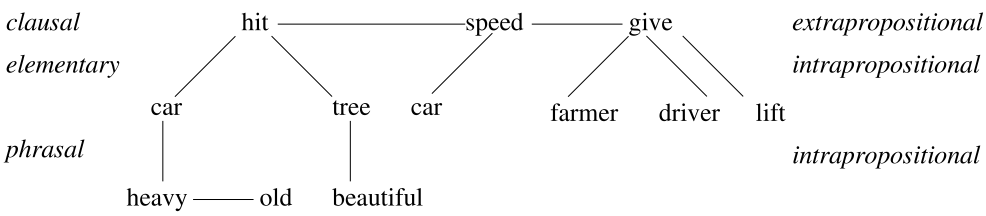
(i) phrase structure grammar *(ii) dependency grammar* *(iii) DBS*



In the (i) PSG graph, the modifier|modified relation is not expressed directly, but must be deduced from the dominance and precedence configuration of the NP, ADJ, and N nodes. In the (ii) DG graph, the nodes the and little are specified indiscriminately as depending on girl and girl is specified as depending on sleep. In a (iii) DBS graph, however, a "|" line is defined to represent the relation between a modifier and a modified. Given that little is an A and girl is an N, the DBS graph in 7.1.2 characterizes the relation between them explicitly as A|N, i.e. as adnominal (7.2.3).

The semantic relations of structure (12.6.2) apply at all three levels of grammatical complexity, i.e. elementary, phrasal, and clausal. Consider the DBS graph analysis of the content representing a short text, namely The heavy old car hit a beautiful tree. The car had been speeding. A farmer gave the driver a lift.[5]

7.1.3 CLAUSAL, ELEMENTARY, AND PHRASAL RELATIONS



The semantic relations are defined between content proplets. At the clausal level, the matrix verbs of the three component propositions are connected as the conjuncts *hit−speed−give* by extrapropositional coordination. The matrix verbs form elementary subject/predicate relations with *car, car,* and *farmer*, respectively, and object\predicate relations with *tree, driver,* and *lift*. The first subject is in an adnominal modifier|modified relation with *heavy|car* at the phrasal level, and the modifier in turn forms the conjunction *heavy−old* (intrapropositional coordination). Also, the adnominal *beautiful* is in a modifier|modified relation with the object *tree* .

[2] In contrast to PSG, Arc Pair grammar (Johnson and Postal 1980), Relational grammar (Perlmutter 1980), and Functional grammar (Dik 1989, 1997), define grammatical functions such as "subject" and "object" as elementary, which resembles the DBS approach to the semantic relations of structure.

[3] We are referring here to DG as originally presented by Tesnière (1959). In more recent work on DG, such as Liu (2009b) and Hudson (2010), dependencies are differentiated by annotating edges.

[4] A differentiated semantic interpretation of graph structures was pioneered by Frege (1879), Begriffsschrift.

[5] See NLC Chap. 13 for the hear mode derivation and Chap. 14 for the corresponding speak mode derivation with explicit applications of the DBS operations.

7.2 Absorption and Precipitation of Function Words

PSG and DG graphs on the one hand and the corresponding DBS graphs on the other differ in the treatment of function words. For example, the definite article the appears as a node in the PSG and DG graphs of 7.1.2, but not in the DBS graph. This is because the PSG and DG graphs analyze language, while the DBS graph analyzes content, at the language and the context level (4.3.3). For a representation of language content it does not matter whether a certain semantic value such as definiteness is coded by means of a function word, as in English, or by means of a suffix, as in Romanian.[6]

Function words of English are determiners like the, a(n), some, every, all (NLC 6.4.6, 6.4.7); auxiliaries like have, had, am, are, were, do, does, did (NLC A.5.4–A.5.6); prepositions like in, on, under (NLC A.4.5); coordinating conjunctions like and, or (NLC Sect. A.7.1), subordinating conjunctions like that, when; and punctuation marks like **.**, **?**, and **!** (NLC A.5.8).

In accordance with surface compositionality, function words are lexically analyzed as proplets. In the hear mode, a function word and its content word are fused and their properties are inherited by the proplet resulting from such a *function word absorption*. For example, the auxiliary was introduces the value past into the resulting verb proplet, the determiner the introduces the value def into the resulting noun proplet, etc.[7] In the speak mode, these same values are used for *function word precipitation*

In function word absorption, values may be absorbed from the content word proplet into the function word proplet or vice versa. For example, in a determiner-noun combination, the relevant values of the content proplet (noun) are copied into the function proplet (determiner), after which the content proplet is discarded. Conversely, in a verb-period combination: the relevant values of the function word proplet (period) are copied into the content proplet (verb), after which the function word proplet is discarded.

Some function words may take more than one syntactic-semantic role:

7.2.1 The function words by and to taking different roles

1a. Mary has a house *by* the lake.
1b. The book was read *by* Mary.

2a. Mary moved *to* Paris.
2b. Mary tried *to* sleep.

[6] For example, om (man) is marked as definite with the suffix -ul, as in om-ul (the man).

[7] Function word absorption has the practical advantage that the high frequency words with low semantic significance (Zipf 1932, 1935, 1949) are moved out of the way of a meaningful retrieval.

Function words in the a-variants combine with the following noun, resulting in a modifier (NLC 15.2.1, 6.6.2–6.6.4). In the b-variants, the same function words combine with the preceding verb, creating an argument position (9.2.2, 15.4.1). Absorbing the function words by and to into a following noun (a-variant) or into a preceding verb (b-variant) is handled by the hear mode operations which combine the function words with their content word. In this way a lexical ambiguity of the function words in question is avoided (surface compositionality).

Because of function words, an expression may be phrasal at the surface level, but elementary at the level of content, as in the following nouns:

7.2.2 Correlating elementary/phrasal surfaces and contents

elementary surface	*phrasal surface*	*phrasal surface*
she	the girl	the little girl
elementary content	*elementary content*	*phrasal content*

$$\begin{bmatrix}\text{noun: pro3}\\ \text{cat: snp}\\ \text{sem: f sg}\\ \text{fnc: sleep}\\ \text{mdr:}\\ \text{prn: 1}\end{bmatrix}\qquad\begin{bmatrix}\text{noun: girl}\\ \text{cat: snp}\\ \text{sem: def f sg}\\ \text{fnc: sleep}\\ \text{mdr:}\\ \text{prn: 2}\end{bmatrix}\qquad\begin{bmatrix}\text{noun: girl}\\ \text{cat: snp}\\ \text{sem: def f sg}\\ \text{fnc: sleep}\\ \text{mdr: little}\\ \text{prn: 3}\end{bmatrix}\begin{bmatrix}\text{noun: little}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd: girl}\\ \text{nc:}\\ \text{prn: 3}\end{bmatrix}$$

While she (elementary surface) and the girl (phrasal surface) are each represented by a single proplet (elementary content), the little girl is represented by two proplets, connected by the prn value 3 and the modifier|modified relation, coded by the mdr and mdd values of the two proplets.

Function word absorption is integrated into the strictly surface compositional, strictly time-linear derivation order of the DBS hear mode:

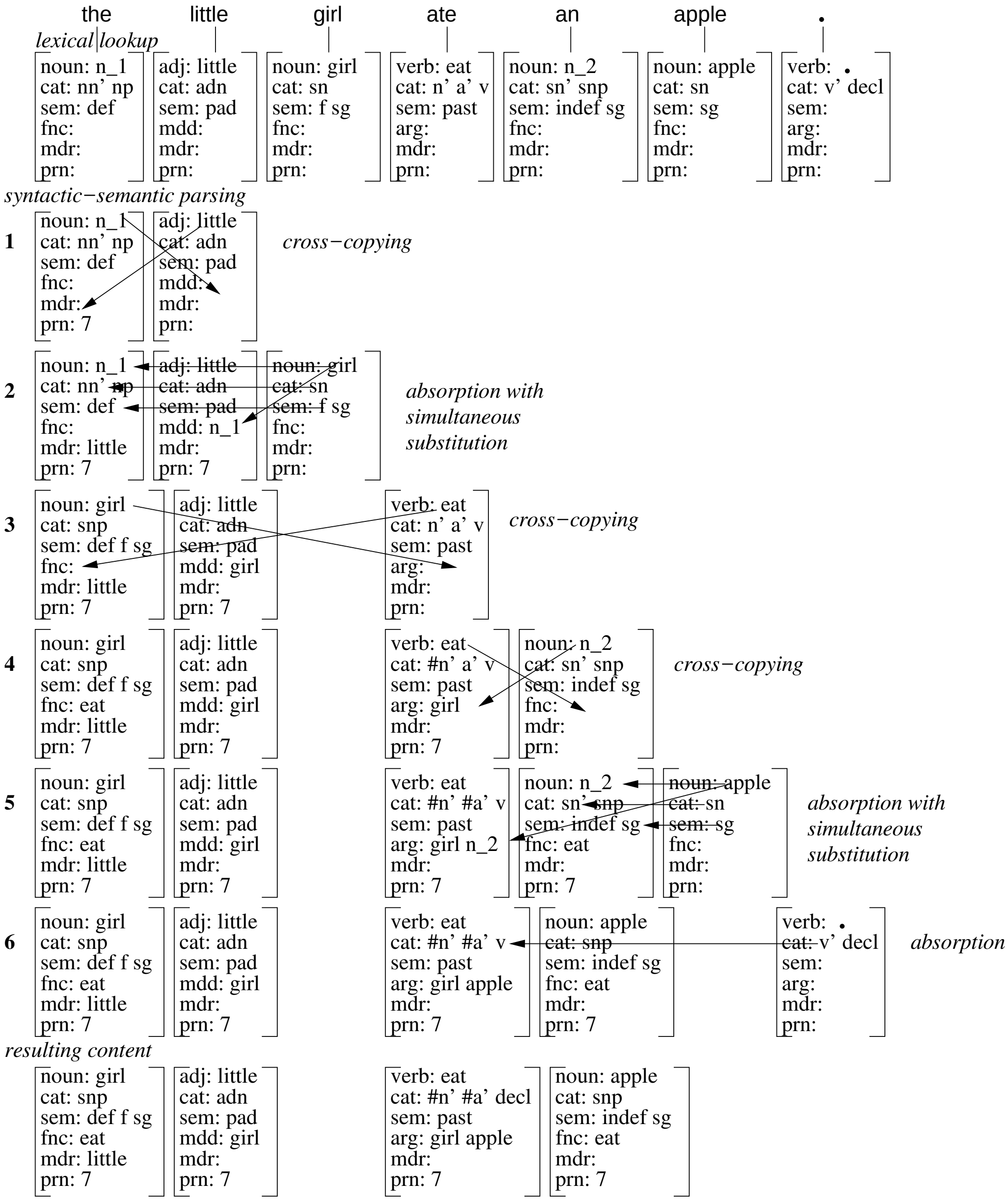
7.2.3 Combining proplets into a phrasal noun

$$\begin{bmatrix}\text{sur: the}\\ \text{noun: n\_1}\\ \text{cat: nn}'\text{ np}\\ \text{sem: def}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 7}\end{bmatrix} + \begin{bmatrix}\text{sur: little}\\ \text{adj: little}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{sur:}\\ \text{noun: n\_1}\\ \text{cat: nn}'\text{ np}\\ \text{sem: def}\\ \text{fnc:}\\ \text{mdr: little}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 7}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{adj: little}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd:n\_1}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 7}\end{bmatrix} + \begin{bmatrix}\text{sur: girl}\\ \text{noun: girl}\\ \text{cat: sn}\\ \text{sem: sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{sur:}\\ \text{noun: girl}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc:}\\ \text{mdr: little}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 7}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{adj: little}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd: girl}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 7}\end{bmatrix}$$

The determiner *the* (function word) first combines with the adj *little* and then absorbs the noun *girl*. The two resulting proplets contain all the necessary information for precipitating the correct surface in the speak mode (7.4.2).

Phrasal nouns in preverbal position are put together first to be combined with the verb as a whole, while postverbal phrasal nouns are attached to the verb step by step. This apparent asymmetry may be illustrated as follows:

7.2.4 HEAR MODE DERIVATION OF The little girl ate an apple.



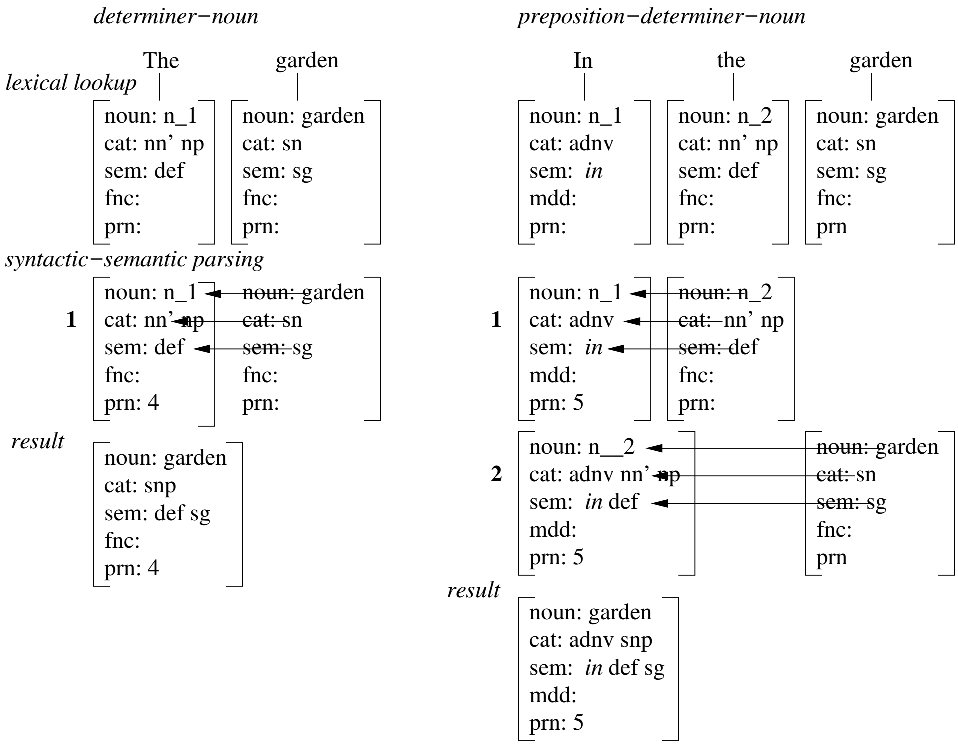
Thanks to variables, pre- and postverbal phrasal nouns are derived by the same operations (NLC Sect. 13.2), despite the asymmetry in the concatenation.

In line 1, the core values of the proplets representing the and little are cross-copied into the respective mdr (modifier) and mdd (modified) slots. In line 2, the core value of *girl* is absorbed into the determiner proplet. In line 3, the core values of the subject and the predicate are cross-copied into the respective fnc and arg slot. In line 4, the core values of the predicate and the (beginning of the) object are cross-copied into the second arg slot of the verb and the fnc slot of the second noun. In line 5, the *apple* proplet is absorbed into the determiner. In line 6, the period is absorbed into the predicate.

The absorption in line 3 involves (i) replacing the two occurrences of the substitution value n_1 (which originated in the lexical determiner proplet) with the core value of the *girl* proplet, (ii) converting the cat values nn′ np in the determiner into snp (singular noun phrase, NLC 13.2.4, 13.3.3), (iii) copying the f sg values into the sem slot of the former determiner proplet, and (iv) discarding the *girl* proplet. The gap resulting from this absorptions begins in lines 3. And similarly for the absorption in line 5.

Another construction with function word absorption is a prepositional phrase like in the garden, compared below with the phrasal noun the garden both as used in sentence-initial position:

7.2.5 Comparing different function word absorptions



On the left, the proplet representing The has the core attribute noun because after absorbing the noun proplet *garden* the result will be a proplet with the core attribute noun. The single step of the determiner-noun derivation (i) substitutes the n_1 value of *the* with the core value of *garden*, (ii) cancels the nn′ position with the sn value, (iii) replaces the np value with snp, (iv) adds the sg value to the sem attribute of the former *the* proplet, and (v) discards the *garden* proplet (NLC 13.3.3).

On the right, step 1 of the time-linear preposition-determiner-noun derivation combines the two lexical function word proplets *in* and *the* into one noun proplet. Thereby the substitution value n_1 in the preposition proplet is replaced with the substitution value n_2 of the determiner proplet, the def value of the determiner proplet is added to the preposition's sem attribute, and the determiner proplet is discarded. Step 2 combines the proplet resulting from step 1 with the lexical *garden* proplet into a single noun proplet. This is based on replacing the n_2 substitution value with the core value of the *garden* proplet, after which the *garden* proplet is discarded.[8] The lexical preposition proplet introduces the core attribute noun (3.5.4) and the continuation attribute mdd (modified). The nature of the preposition, here *in*, is coded into the first slot of the sem attribute.[9]

Graphically, the adnominal use of a prepositional phrase is represented as N|N, while the adverbial use is represented as N|V. The corresponding constructions with an elementary adjective are A|N and A|V. An elementary adverbial, e.g. Julia slept there. and a phrasal adverbial, e.g. Julia slept in the garden. differ graphically as follows (and accordingly for adnominals, cf. 7.3.6; NLC Sects. 7.3, 7.4):

7.2.6 Elementary adjective vs. prepositional phrase

elementary: A|V *phrasal*: N|V

The elementary modifier|modified is A|V, while the phrasal counterpart is N|V. There is also the distinction between N/V (intrapropositional) and V/V

[8] For a more detailed discussion of prepositional phrases see NLC 6.5.4 f., Sects. 15.1–15.4.

[9] As shown in 3.5.5, DBS uses the cat value adn (adnominal) for elementary modifiers restricted morphologically to nouns, e.g. beautiful, adv (adverbial) for elementary modifiers restricted morphologically to verbs, e.g. beautifully, and adnv for elementary modifiers which may be applied to verbs or nouns, e.g. fast. Because all prepositional phrases may be used adnominally and adverbially, they use adnv as their cat value as well. Phrasal and elementary adnvs differ in their core attribute.

(clausal) subject, $\mathrm{N}\backslash\mathrm{V}$ (intrapropositional) and $\mathrm{V}\backslash\mathrm{V}$ (clausal) object, $\mathrm{A}|\mathrm{N}$ (intrapropositional) and $\mathrm{V}|\mathrm{N}$ (clausal) adnominal, $\mathrm{A}|\mathrm{V}$ (intrapropositional) and $\mathrm{V}|\mathrm{V}$ (clausal) adverbial, and $\mathrm{N}{-}\mathrm{N}$, $\mathrm{V}{-}\mathrm{V}$, $\mathrm{A}{-}\mathrm{A}$ (intrapropositional) and $\mathrm{V}{-_x}\mathrm{V}$ (clausal) conjunct.

7.3 Deriving DBS Graphs from Sets of Proplets

The graphs (ia) and (iia) in 7.2.6 are called *semantic relations graphs*, or *SRG*s for short, while the graphs (ib) and (iib) are called *signatures*. An SRG resembles a DG graph insofar as it uses English words, while a signature resembles a phrase structure tree insofar as it uses letters for nodes instead (7.1.1, 7.1.2). Defined as conceptual superstructures on top of sets of proplets, SRGs and signatures characterize a content graphically in terms of its semantic relations of structure.

A signature is more general than an SRG because it abstracts away from the content proplets by replacing them with letters representing their core *attribute*. An SRG, in contrast, is more intuitive in that it illustrates a content construction concretely with the core *values*,[10] using English words as place holders for procedural implementations (grounding).

For the software procedures of DBS, the nodes of the SRG or the signature must be replaced by proplets (7.3.2). This raises the following questions:

7.3.1 On relating proplet sets to DBS graphs

- What is the nature of the relation between the proplet representation of a content and the corresponding SRG or signature? (Sect. 7.5)
- Is it possible to derive an SRG or a signature automatically from a set of concatenated proplets which represent a content?

Consider the content derived in 7.2.4 as an order-free set of proplets:[11]

7.3.2 Content corresponding to The little girl ate an apple.

$$\begin{bmatrix}\text{noun: girl}\\ \text{cat: snp}\\ \text{sem: def f sg}\\ \text{fnc: eat}\\ \text{mdr: little}\\ \text{prn: 7}\end{bmatrix}\begin{bmatrix}\text{adj: little}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd: girl}\\ \text{mdr:}\\ \text{prn: 7}\end{bmatrix}\begin{bmatrix}\text{verb: eat}\\ \text{cat: \#n}'\text{ \#a}'\text{ decl}\\ \text{sem: past}\\ \text{arg: girl apple}\\ \text{mdr:}\\ \text{prn: 7}\end{bmatrix}\begin{bmatrix}\text{noun: apple}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: eat}\\ \text{mdr:}\\ \text{prn: 7}\end{bmatrix}$$

[10] As pointed out by Haitao Liu, an SRG resembles the stemma réelle while a signature resembles the stemma virtuelle of Tesnière (1959). See also Ágel et al. (eds.) (2006), p. 1316.

[11] Preparing a speak mode derivation in the corresponding hear mode derivation is in line with the linguistic laboratory setup of DBS (Sect. 14.4; NLC 3.5.2; TExer Sect 1.5.

This set of proplets may be mapped automatically into an SRG or a signature by forming all possible unordered pairs of related proplets, i.e. {*girl, eat*}, {*apple, eat*}, and {*little, girl*}, and by applying the following patterns to each:

7.3.3 Patterns interpreting intrapropositional relations

(1) noun/verb
$$\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } \beta\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg: } \alpha \text{ X}\end{bmatrix}$$

(2) noun\verb
$$\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } \beta\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg: } \gamma \text{ X } \alpha\end{bmatrix}$$

(3) adjective|noun
$$\begin{bmatrix}\text{adj: } \alpha \\ \text{mdd: } \beta\end{bmatrix} \begin{bmatrix}\text{noun: } \beta \\ \text{mdr: } \alpha\end{bmatrix}$$

(4) adjective|verb
$$\begin{bmatrix}\text{adj: } \alpha \\ \text{mdd: } \beta\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{mdr: } \alpha\end{bmatrix}$$

(5) noun− noun
$$\begin{bmatrix}\text{noun: } \alpha \\ \text{nc: } \beta\end{bmatrix} \begin{bmatrix}\text{noun: } \beta \\ \text{pc: } \alpha\end{bmatrix}$$

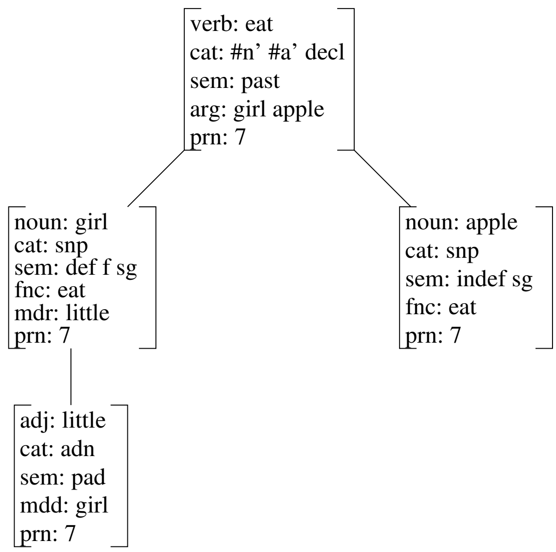
(6) verb−verb
$$\begin{bmatrix}\text{verb: } \alpha \\ \text{nc: } \beta\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{pc: } \alpha\end{bmatrix}$$

(7) adjective−adjective
$$\begin{bmatrix}\text{adj: } \alpha \\ \text{nc: } \beta\end{bmatrix} \begin{bmatrix}\text{adj: } \beta \\ \text{pc: } \alpha\end{bmatrix}$$

The only pair matching pattern (1) is {*girl, eat*}, the only pair matching pattern (2) is {*apple, eat*}, and the only pair matching pattern (3) is {*little, girl*}.

Arranging and connecting the proplets in accordance with the successful matches between 7.3.2 and 7.3.3 results in the following graph:

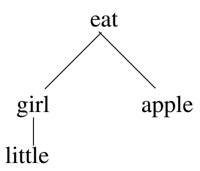
7.3.4 DBS graph based on proplets (proplet graph)



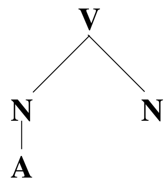
In a second step, this proplet graph may be turned automatically into an SRG or a signature. The SRG is obtained by replacing each proplet with its core *value*, while the signature is obtained by replacing each proplet with a letter representing its core *attribute*:

7.3.5 Resulting SRG and signature

(i) *SRG (semantic relations graph)*



(ii) *signature*



The SRG and the signature are each *homomorph*[12] to the proplet graph 7.3.4. The direction of the homomorphism from SRGs and signatures to proplet graphs is necessitated by the more differentiated structure of proplets as compared to the corresponding nodes in SRGs or signatures.

The complex signature in 7.3.5 is composed of the elementary signatures N/V, A|N, and N\V. In each, the first node is the lower one in the graph and the second node is the higher one. In "−" signatures (coordination) the first node is shown in the graph as preceding the second node (9.6.2).

At the elementary and phrasal level of grammatical complexity, there are the following elementary signatures in English (in linear notation):

7.3.6 Seven intrapropositional semantic relations

subject/predicate:	1. N/V
object\predicate:	2. N\V
adjective|noun:	3. A|N
adjective|verb:	4. A|V
conjunct−conjunct:	5. N−N
	6. V−V
	7. A−A

Corresponding extrapropositional signatures are presented in Sect. 7.6.

7.4 Producing Natural Language Surfaces from Content

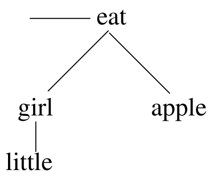
For the production of natural language from content (speak mode), the static (i) SRG and (ii) signature are complemented by two additional structures which are dynamic and called the (iii) *numbered arcs graph* or NAG[13] and the (iv) *surface realization*.

[12] Structure A is homomorph to a structure B iff (i) each elementary node in A has a counterpart in B and (ii) each relation in A has a corresponding relation in B (FoCL Sect. 21.3).

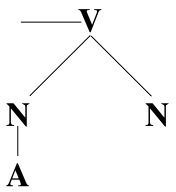
[13] For graph-theoretical and linguistic constraints on NAGs see Sect. 9.1.

7.4.1 THE FOUR DBS VIEWS ON A CONTENT AND ITS SURFACE

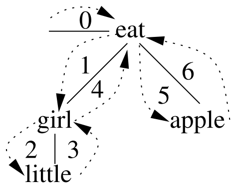
(i) *SRG (semantic relations graph)*



(ii) *signature*



(iii) *NAG (numbered arcs graph)*



(iv) *surface realization*

1	2	3	4	5	6
The	little	girl	ate	an_apple	.
V/N	N|A	A|N	N/V	V\N	N\V

Together, these conceptual structures provide four simultaneous views (in the conceptual sense) on a given content.

The (i) SRG and the (ii) signature are the same[14] as in 7.3.5. The numbering of arcs in the (iii) NAG is known as *depth first* (pre-order)[15] in graph theory.

The (iv) surface realization consists of three lines. The numbers of the top line refer to the arcs in the NAG; they show which surface in the middle line is realized in which arc. The bottom line shows for each traversal the name of the DBS.Nav operation[16] which drives the time-linear navigation (7.4.3).

In the static view of the SRG and the signature, a semantic relation between two nodes is represented by a single line, e.g. N/V. Thereby the lower node precedes.

In the dynamic view of the NAG, however, this single line is traversed in two directions. In linear notation, they are written as, e.g. V↙N for moving from the predicate to the subject (arc 1) and N↗V for moving from the subject back to the predicate (arc 4). Thereby the respective first node is the point of origin and the second node the goal.

The surface realization is provided by language-dependent lexicalization rules which are embedded into the sur slot of the goal proplet of a speak mode operation (3.4.3). In 7.4.1, traversal of arc 1 realizes The from the sem value def in *girl*, arc 2 little from the core value of *little*, arc 3 girl from the core

[14] Except for the additional "start line" (arc 0). Cf. 9.2.5 f., 9.3.4, 9.3.5.

[15] The pre-order numbering of an intrapropositional NAG starts and ends at the root, here the top verb. It traverses all nodes in a top down, left to right fashion, and returns right to left bottom up.

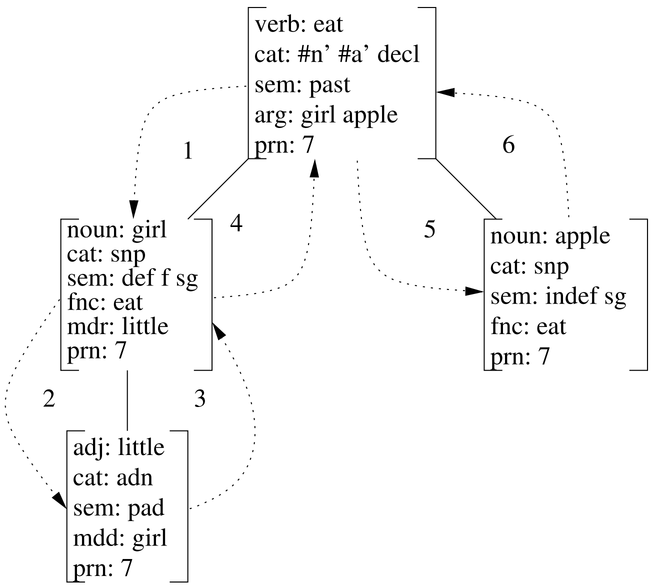
[16] Because the graphics software used for the surface realization does not provide for the arrow heads in, e.g. V↙N, it is written as V/N and the inverse arc N↗V as N/V. Thus, the interpretation of, e.g. N/V, depends on whether it refers to a semantic relation in a signature or to an arc in a NAG.

value and the sem value sg of *girl*, arc 4 ate from the core and the sem value past of *eat*, arc 5 an_apple (multiple realization) from the sem and the core value of *apple*, and arc 6 the period from the cat value decl of *eat*.

A numbered arcs graph may be defined for an SRG, a signature, or a proplet graph. Consider the NAG corresponding to the English surface The little girl ate an apple., familiar from 7.2.4, 7.3.2, 7.3.4 and 7.3.5:

7.4.2 Numbered arcs graph based on proplets (proplet NAG)

proplet NAG



Given this proplet NAG, defining the DBS.Speak operations for English is easy: 1. take the proplet pair (elementary signature) of the traversal in question, 2. define the operation patterns by (i) selecting the relevant attributes and (ii) replacing their constant values with variables, and 3. derive a surface from certain proplet values using the appropriate lexicalization rule as defined in NLC, Sects. 12.4–12.6. For the proplet NAG 7.4.2, this procedure results in the definition of the following DBS.Speak grammar:

7.4.3 DBS.Speak grammar for The little girl ate an apple.

An intrapropositional navigation begins with the top verb. Subsequent traversals are determined by the input available to the operations.

$$
\begin{array}{lccl}
 & \text{V}\swarrow\text{N} & & \\
\textit{pattern level:} & \begin{bmatrix}\text{verb: } \alpha \\ \text{arg: } \beta\ \text{X} \\ \text{prn: K}\end{bmatrix} \Rightarrow & \begin{bmatrix}\text{sur: lexnoun}(\hat{\beta}) \\ \text{noun: } \beta \\ \text{fnc: } \alpha \\ \text{prn: K}\end{bmatrix} & \text{\#-mark } \beta \text{ in the } \mathsf{arg} \text{ slot of proplet } \alpha \\
 & \Uparrow & \Downarrow & \\
\textit{content level:} & \begin{bmatrix}\text{verb: eat} \\ \text{cat: \#n}'\ \text{\#a}'\ \text{decl} \\ \text{sem: past} \\ \text{arg: girl apple} \\ \text{prn: 7}\end{bmatrix} & \begin{bmatrix}\text{sur: The} \\ \text{noun: girl} \\ \text{cat: snp} \\ \text{sem: def f sg} \\ \text{fnc: eat} \\ \text{mdr: little} \\ \text{prn: 7}\end{bmatrix} &
\end{array}
$$

Comment: This operation application corresponds to traversal of arc 1 in the proplet NAG 7.4.2. Using the first value of the arg attribute, V/N navigates to the *girl* proplet. The lexicalization rule *lexnoun* (NLC 12.5.2) realizes the surface The. To prevent a relapse or a loop, the first value girl in the arg slot of *eat* is marked with # by the *instruction* shown to right of the operation pattern. Of the operations beginning with an N pattern (NLC Sect. 12.2), N|A applies next:

$$
\begin{array}{lccl}
 & \textbf{N}\downarrow\textbf{A} & & \\
\textit{pattern level} & \begin{bmatrix}\text{noun: } \beta \\ \text{mdr: } \gamma \\ \text{prn: K}\end{bmatrix} \Rightarrow & \begin{bmatrix}\text{sur: lexadj}(\hat{\gamma}) \\ \text{adj: } \gamma \\ \text{mdd: } \beta \\ \text{prn: K}\end{bmatrix} & \text{\#-mark } \gamma \text{ in the } \mathsf{mdr} \text{ slot of proplet } \beta \\
 & \Uparrow & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{noun: girl} \\ \text{cat: snp} \\ \text{sem: def f sg} \\ \text{fnc: eat} \\ \text{mdr: little} \\ \text{prn: 7}\end{bmatrix} & \begin{bmatrix}\text{sur: little} \\ \text{adj: little} \\ \text{cat: adn} \\ \text{sem: pad} \\ \text{mdd: girl} \\ \text{prn: 7}\end{bmatrix} &
\end{array}
$$

Comment: This operation application corresponds to the traversal of arc 2 in the proplet NAG 7.4.2. By matching and binding the variable γ of the N pattern to the mdr value little, the goal proplet A is accessed and used by the function *lexadj* (NLC 12.6.1) to realize the surface little. Here, the instruction #-marks the mdr value little.

The application of N↓A is followed by A↑N, which uses the mdd value girl as its continuation value:

$$
\begin{array}{lccl}
 & \textbf{A}\uparrow\textbf{N} & & \\
\textit{pattern level} & \begin{bmatrix}\text{adj: } \gamma \\ \text{mdd: } \beta \\ \text{nc: Z} \\ \text{prn: K}\end{bmatrix} \Rightarrow & \begin{bmatrix}\text{sur: lexnoun}(\hat{\beta}) \\ \text{noun: } \beta \\ \text{mdr: \#}\gamma \\ \text{prn: K}\end{bmatrix} & \text{\#-mark } \beta \text{ in the } \mathsf{mdd} \text{ slot of proplet } \gamma \\
 & \text{where Z is \#-marked or NIL} & & \\
 & \Uparrow & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{adj: little} \\ \text{cat: adn} \\ \text{sem: pad} \\ \text{mdd: girl} \\ \text{prn: 7}\end{bmatrix} & \begin{bmatrix}\text{sur: girl} \\ \text{noun: girl} \\ \text{cat: snp} \\ \text{sem: def f sg} \\ \text{fnc: eat} \\ \text{mdr: \#little} \\ \text{prn: 7}\end{bmatrix} &
\end{array}
$$

Comment: This operation application corresponds to the traversal of arc 3 in the proplet NAG 7.4.2. The lexicalization rule *lexnoun* (NLC 12.5.2) realizes the surface girl, completing the phrasal noun. The application of A↑N is followed by N↗V, which uses the fnc value eat as its continuation value:

N↗V

$$
\begin{array}{llcll}
\textit{pattern level} & \begin{bmatrix}\text{noun: } \beta \\ \text{fnc: } \alpha \\ \text{mdr: Z} \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{sur: lexverb}(\hat{\alpha}) \\ \text{verb: } \alpha \\ \text{arg: \#}\beta \text{ Y} \\ \text{prn: K}\end{bmatrix} & \text{\#-mark } \alpha \text{ in the } \mathsf{fnc} \text{ slot of proplet } \beta \\
 & \text{where Z is \#-marked or NIL} & & & \\
 & \Uparrow & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{noun: girl} \\ \text{cat: snp} \\ \text{sem: def f sg} \\ \text{fnc: eat} \\ \text{mdr: \#little} \\ \text{prn: 7}\end{bmatrix} & & \begin{bmatrix}\text{sur: } \mathsf{ate} \\ \text{verb: eat} \\ \text{cat: \#n}' \text{ \#a}' \text{ decl} \\ \text{sem: past} \\ \text{arg: \#girl apple} \\ \text{prn: 7}\end{bmatrix} &
\end{array}
$$

Comment: This operation application corresponds to the traversal of arc 4 in the proplet NAG 7.4.2. N↗V moves from the subject noun back to the verb. The subject value of the verb's arg attribute has already been marked with # in the derivation-initial application of V↙N. Based on the sem value past, *lexverb* realizes the surface ate from the core value of the V proplet.

Of the operations beginning with a V pattern, V−V (extrapropositional coordination) fails because there is still an unmarked value in the arg slot of *eat*, namely apple. V↘N, however, applies as follows:

V↘N

$$
\begin{array}{llcll}
\textit{pattern level} & \begin{bmatrix}\text{verb: } \alpha \\ \text{arg: \#X } \beta \text{ Y} \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{sur: lexnoun}(\hat{\beta}) \\ \text{noun: } \beta \\ \text{fnc: \#}\alpha \\ \text{prn: K}\end{bmatrix} & \text{\#-mark } \beta \text{ in the } \mathsf{arg} \text{ slot of proplet } \alpha \\
 & \Uparrow & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{verb: eat} \\ \text{cat: \#n}' \text{ \#a}' \text{ decl} \\ \text{sem: past} \\ \text{arg: \#girl apple} \\ \text{prn: 7}\end{bmatrix} & & \begin{bmatrix}\text{sur: } \mathsf{an\_apple} \\ \text{noun: apple} \\ \text{sem: indef sg} \\ \text{fnc: eat} \\ \text{mdr: } \\ \text{prn: 7}\end{bmatrix} &
\end{array}
$$

Comment: This operation application corresponds to the traversal of arc 5 in the proplet NAG 7.4.2. Given that the goal proplet N is not modified, the rule *lexnoun* applies and realizes the surface an_apple.

Finally the navigation returns to the verb to realize the period (arc 6 in the proplet NAG 7.4.2).

N↖V

$$
\begin{array}{llcl}
\textit{pattern level} & \begin{bmatrix}\text{noun: } \beta \\ \text{fnc: \#}\alpha \\ \text{mdr: Z} \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{lexverb}(\hat{\alpha}) \\ \text{verb: } \alpha \\ \text{arg: \#X \#}\beta \text{ Y} \\ \text{prn: K}\end{bmatrix} \\
 & \text{where Z is \#-marked or NIL} & & \\
 & \Uparrow & & \Downarrow \\
\textit{content level} & \begin{bmatrix}\text{noun: apple} \\ \text{cat: snp} \\ \text{sem: def sg} \\ \text{fnc: eat} \\ \text{mdr: } \\ \text{prn: 7}\end{bmatrix} & & \begin{bmatrix}\text{sur: } \bullet \\ \text{verb: eat} \\ \text{cat: \#n}' \text{ \#a}' \text{ decl} \\ \text{sem: past} \\ \text{arg: \#girl \#apple} \\ \text{prn: 7}\end{bmatrix}
\end{array}
$$

Comment: With all arg values of the verb #-marked (canceled), the navigation may continue with V→V (extrapropositional coordination) to the top verb of a possible next proposition (provided the nc slot of the input has an extrapropositional value).

The translation of DBS.Speak operations, specified in the manner shown above, into the JSLim software is routine.[17]

7.5 Semantic Relations of Structure

For characterizing the semantic relations of structure in nonlanguage contents (and, by extension, in natural language contents), DBS uses (i) the short graphical notation of elementary signatures, e.g. N/V, and (ii) a more detailed representation as pairs of proplets. The latter serves as the declarative specification for the DBS software. The following examples explain the relation between these two kinds of representation, going systematically through the main semantic relations of structure listed in 7.6.4 and 7.6.5.

The basic principle of connecting two proplets semantically is using the same values in different slots. Let us begin with the proplet analysis of the intrapropositional subject/predicate relation N/V, as in The man gave... (NLC 6.1.1, 6.1.3, 6.1.4).

7.5.1 INTRAPROPOSITIONAL SUBJECT/PREDICATE (N/V)

$$\begin{bmatrix}\text{verb: give}\\ \text{arg: man child apple}\\ \text{prn: 1}\end{bmatrix}$$

$$\begin{bmatrix}\text{noun: man}\\ \text{fnc: give}\\ \text{prn: 1}\end{bmatrix}$$

The value give appears in the verb slot of the first proplet and the fnc slot of the second. The value man appears in the noun slot of the second proplet and the first arg slot of the first. The orientation of the dotted lines corresponds to V↙N (downward arc) and N↗V (upward arc) in the associated NAG.

In other words, the pair of proplets is connected semantically by two *corresponding value pairs* which establish a duplex relation (16.6.7). The two directions of the relation are coded by means of addresses which consist of a core and a prn value. The use of addresses makes proplets order-free (3.2.7), i.e. the semantic relations of structure are maintained regardless of where the proplets are stored in accordance with the database schema used.

The subject proplet N in 7.5.1 may have additional continuation features such as mdr or nc (not shown), but they would not affect the N/V relation.

[17] NLC Chaps. 12–14 present DBS.Hear and DBS.Speak grammars for two small texts. TExer presents a DBS.Hear and a DBS.Speak grammar for 24 constructions of increasing grammatical complexity.

Similarly for the predicate proplet V: if it happened to be the continuation value of other (higher) proplets (NLC Chap. 7), as in a clausal relation, this would have no consequence on the strictly local N/V relation at hand.

In the corresponding extrapropositional (clausal) subject/predicate relation (V/V), the lower verb has a double function: it serves (i) as the predicate of the subject clause and (ii) as the subject of the higher verb (such as barked in That Fido barked amused Mary, NLC 7.1.1–7.1.4):

7.5.2 Extrapropositional subject/predicate (V/V)

$$\begin{bmatrix} \text{verb: amuse} \\ \text{arg: (bark 2) Mary} \\ \text{prn: 1} \end{bmatrix}$$

$$\begin{bmatrix} \text{verb: bark} \\ \text{arg: Fido} \\ \text{fnc: (amuse 1)} \\ \text{prn: 2} \end{bmatrix}$$

The crucial properties of an extrapropositional subject/predicate relation (as compared to the corresponding intrapropositional relation 7.5.1) are (i) the different prn values of the two propositions, (ii) the core attribute verb (rather than noun) of the proplet representing the subject, (iii) its additional fnc attribute, introduced lexically by the subordinating conjunction that (NLC A.4.4) and needed for the connection to the higher clause, (iv) the long (extrapropositional) address value of the subject's fnc attribute, and (v) the long address value in the initial arg slot of the predicate.

Next let us consider the object\predicate relation. It requires a verb with more than one arg slot (transitive): while the subject/predicate relation is established between the subject and the *initial* arg value of the predicate, the object\predicate relation is established between an object and a *non-initial* arg value of the predicate. The following example shows the intrapropositional object\predicate relation in ... gave the child... (NLC 6.1.1–6.1.4).

7.5.3 Intrapropositional object\predicate (N\V)

$$\begin{bmatrix} \text{verb: give} \\ \text{arg: man child apple} \\ \text{prn: 1} \end{bmatrix}$$

$$\begin{bmatrix} \text{noun: child} \\ \text{fnc: give} \\ \text{prn: 1} \end{bmatrix}$$

The intrapropositional nature of the relation is coded by the two proplets having the same prn value and the use of short address values. The orientation of the dotted lines corresponds to $V \searrow N$ (downward arc) and $N \nwarrow V$ (upward arc) in the associated NAG.

The distinction between intra- and extrapropositional object\predicate relations is similar to that of subject/predicate. In extrapropositional object\predicate relations, (i) there are two different prn values, the object proplet has (ii) the core attribute verb and (iii) an additional fnc attribute, introduced lexically by the subordinating conjunction *that* (NLC A.4.4) and needed for connecting the verb of the subclause to the higher verb. Also, (iv) the fnc attribute of the object and (v) the corresponding non-initial arg value of the predicate take long address values (see NLC Sect. 7.2 for Mary heard that Fido barked):

7.5.4 EXTRAPROPOSITIONAL OBJECT\PREDICATE (V\V)

$$\begin{bmatrix}\text{verb: hear}\\ \text{arg: Mary (bark 2)}\\ \text{prn: 1}\end{bmatrix}$$

$$\begin{bmatrix}\text{verb: bark}\\ \text{arg: Fido}\\ \text{fnc: (hear 1)}\\ \text{prn: 2}\end{bmatrix}$$

While the intrapropositional subject/predicate (7.5.1) and object\predicate (7.5.3) relations are defined (i) between the verb value of the predicate and fnc value of the argument, and (ii) between the noun value of the argument and an arg value of the predicate, an intrapropositional adnominal|noun relation is defined (i) between the noun value of the modified and the mdd value of the modifier, and (ii) between the adj value of the modifier and the mdr value of the modified (as in ... little dog, NLC Sect. 6.2):

7.5.5 INTRAPROPOSITIONAL ADNOMINAL|NOUN (A|N)

$$\begin{bmatrix}\text{noun: dog}\\ \text{mdr: little}\\ \text{prn:1}\end{bmatrix}$$

$$\begin{bmatrix}\text{adj: little}\\ \text{mdd: dog}\\ \text{prn: 1}\end{bmatrix}$$

The corresponding value pairs are (i) dog in the noun slot of the modified and the mdd slot of the modifier, and (ii) little in mdr slot of the modified and the adj slot of the modifier. As in 7.5.1 and 7.5.3, the intrapropositional nature of the relation is coded by the two proplets having the same prn value and the use of short address values. The orientation of the dotted lines corresponds to $N \downarrow A$ (downward arc) and $A \uparrow N$ (upward arc) in the associated NAG.

The extrapropositional counterpart of the adnominal|noun relation is also known as the relation between a "relative clause" and its "head noun." In DBS, it is defined between a verb proplet representing the modifier clause and a noun proplet representing the modified. Thereby, the core value of the noun is shown as a gap (indicated by ∅) in the arg slot of the verb, as in The dog which saw Mary barked (subject gap, NLC Sect. 7.3) and The dog which Mary saw barked (object gap, NLC Sect. 7.4).

7.5.6 Extrapropositional adnominal|noun (V|N)

$$\begin{bmatrix}\text{noun: dog}\\ \text{fnc: bark}\\ \text{mdr: (see 2)}\\ \text{prn: 1}\end{bmatrix}$$

$$\begin{bmatrix}\text{verb: see}\\ \text{arg: } \emptyset \text{ Mary}\\ \text{mdd: (dog 1)}\\ \text{prn: 2}\end{bmatrix}$$

The lower verb has an additional mdd attribute, introduced lexically by the subordinating conjunction *which* (NLC A.6.3) and needed for the concatenation with the modified noun. The corresponding value pairs are (i) dog in the noun slot of the modified and (dog 1) in the additional mdd slot of the clausal modifier, and (ii) see in the verb slot of the modifier and (see 2) in mdr slot of the modified. The implicit value of the head noun, represented by ∅ in the arg slot of the subclause verb, is specified by the mdd value directly underneath. As in 7.5.2 and 7.5.4, the extrapropositional nature of the relation is coded by the two proplets having different prn values and the use of long addresses.

A fourth functor-argument relation is between a modifier and a predicate. The intrapropositional adverbial|verb relation is defined (i) between the verb value of the modified and the mdd value of modifier, and (ii) the mdd value of the modifier and the verb value of the modified (as in sleep soundly, NLC Sect. 6.3).

7.5.7 Intrapropositional adverbial|verb (A|V)

$$\begin{bmatrix}\text{verb: sleep}\\ \text{mdr: sound}\\ \text{prn: 1}\end{bmatrix}$$

$$\begin{bmatrix}\text{adj: sound}\\ \text{mdd: sleep}\\ \text{prn: 1}\end{bmatrix}$$

The corresponding value pairs are (i) sleep in the verb slot of the modified and the mdd slot of the modifier, and (ii) sound in adj slot of the modifier and the mdr slot of the modified.

The differences between intra- and extrapropositional adverbials resemble those of the other functor-argument relations: the clausal modifier is represented by a verb proplet with an additional attribute, here mdd, introduced lexically by the subordinating conjunction (NLC A.6.4). It serves to connect the modifier clause to the modified verb (as in When Fido barked, Mary laughed, NLC Sect. 7.5). The position of intra- and extrapropositional modifiers is comparatively order-free, even in English (NLC 6.3.1, 7.5.1, 7.5.4, and 15.4.4).

7.5.8 Extrapropositional adverbial|verb (V|V)

$$\begin{bmatrix} \text{verb: laugh} \\ \text{arg: Mary} \\ \text{mdr: (bark 2)} \\ \text{prn: 1} \end{bmatrix}$$

$$\begin{bmatrix} \text{verb: bark} \\ \text{arg: Fido} \\ \text{mdd: (laugh 1)} \\ \text{prn: 2} \end{bmatrix}$$

The corresponding value pairs are (i) laugh in the verb slot of the modified and (laugh 1) in the mdd slot of the modifier, and (ii) bark in verb slot of the modifier and (bark 2) in the mdr slot of the modified.

While the functor-argument relations hold between different kinds of proplets, the conjunct−conjunct relations of coordination hold between the same kinds of proplets. Their concatenation relies on the continuation attributes nc (next conjunct) and pc (previous conjunct), in contradistinction to functor-arguments, which rely on the continuation attributes fnc, arg, mdr, and mdd.

Leaving a coordination of function words such as prepositions, for example above, below, and behind, aside, let us turn to the coordination of content proplets, beginning with noun−noun. The corresponding value pairs are (i) between the core value of the first and the pc value of the second conjunct, and (ii) the core value of the second and the nc value of the first conjunct, as in the man, the woman (NLC 8.1.3):

7.5.9 INTRAPROPOSITIONAL NOUN COORDINATION (N—N)

$$\begin{bmatrix}\text{noun: man\&}\\ \text{fnc: sleep}\\ \text{nc: woman}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix}\begin{bmatrix}\text{noun: woman}\\ \text{fnc:}\\ \text{nc:}\\ \text{pc: man\&}\\ \text{prn: 1}\end{bmatrix}$$

To show "−" relation of coordination, the two proplets are presented as a horizontal sequence, in contradistinction to the functor-argument relations, i.e. "/" in 7.5.1 and 7.5.2, "\" in 7.5.3 and 7.5.4, and "|" in 7.5.5 and 7.5.6. Placing proplets in functor-argument vs. coordination relations differently is for explanation only and should not detract from their order-free nature.

Noun coordinations are integrated into a proposition's functor-argument by &-marking[18] all occurrences of the first conjunct's core value, here man&, and using it as the sole representative of the coordination in the arg slot of the verb (NLC 8.1.4). This method is used with all intrapropositional coordinations, such as adj−adj.

Adj-adj has two uses: (i) adnominal, as in smart, pretty (NLC 8.3.5), and (ii) adverbial, as in slowly, gently (NLC 8.3.6). These uses may be illustrated as follows:

7.5.10 INTRAPROPOSITIONAL ADN AND ADV COORDINATIONS (A—A)

$$\begin{bmatrix}\text{adj: smart\&}\\ \text{mdd: girl}\\ \text{nc: pretty}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix}\begin{bmatrix}\text{adj: pretty}\\ \text{mdd:}\\ \text{nc:}\\ \text{pc: smart\&}\\ \text{prn: 1}\end{bmatrix}\qquad\qquad\begin{bmatrix}\text{adj: slow\&}\\ \text{mdd: talk}\\ \text{nc: gentle}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix}\begin{bmatrix}\text{adj: gentle}\\ \text{mdd:}\\ \text{nc:}\\ \text{pc: slow\&}\\ \text{prn: 1}\end{bmatrix}$$

The corresponding value pairs in the adnominal example on the left are (i) smart& in the adj slot of the first and the pc slot in the second conjunct, and (ii) pretty in the adj slot of the second and the nc slot of the first conjunct. And accordingly for the adverbial example on the right.

An intrapropositional verb coordination resembles intrapropositional noun and adj coordinations, except that it connects verb proplets, as in buy, cook (NLC 8.3.4):

[18] The &-marking of the initial conjunct is necessary because the storage of proplets in the agent's word bank is order-free. Once the initial conjunct is found, the order of the non-initial conjuncts is determined by the nc and pc values (NLC Sect. 8.1).

7.5.11 Intrapropositional verb coordination (V—V)

$$\begin{bmatrix}\text{verb: buy\&} \\ \text{arg: man pizza} \\ \text{nc: cook} \\ \text{pc:} \\ \text{prn: 1}\end{bmatrix} \begin{bmatrix}\text{verb: cook} \\ \text{arg:} \\ \text{nc:} \\ \text{pc: buy\&} \\ \text{prn: 1}\end{bmatrix}$$

Intrapropositional noun, verb, and adj coordinations have in common that (i) any adjacent conjuncts have the same core attribute and the same prn value, (ii) short address values are used in the nc and pc slots, and (iii) the core value of the initial conjunct is &-marked (NLC 8.1.3, 8.1.4).

Because extrapropositional relations require at least one clause represented by a verb proplet (including adnominal modifiers, 7.5.6), extrapropositional coordination is limited to the concatenation of verb proplets which represent clauses (NLC Sect. 9.2). In other words, noun—noun or adj—adj coordinations cannot be extrapropositional.

7.5.12 Extrapropositional verb coordination (V—V)

$$\begin{bmatrix}\text{verb: sleep} \\ \text{nc: (sing 2)} \\ \text{pc:} \\ \text{prn: 1}\end{bmatrix} \quad \begin{bmatrix}\text{verb: sing} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 2}\end{bmatrix}$$

The extrapropositional nature of the relation is coded by the two verb proplets having different prn values and the use of a long address in the nc slot of the first verb. If an extrapropositional coordination is not integrated into a functor-argument[19] there is no need for &-marking the initial conjunct. Extrapropositional coordination in the speak mode is special in using only a simplex connection (16.6.7), here from *sleep* to *sing*. Because extrapropositional V—V relations to not provide pc values, an anti-temporal traversal of coordinated propositions is based on inferencing (Sects. 5.2, 5.5).

7.6 Possible and Actual Semantic Relations of Structure

Intrapropositionally, the number of elementary signatures consisting of the nodes N, V, and A and the lines "/," "\\," "|," and "—" is 36. How many of these are actually used in English? As an initial, tentative approach, consider the following three lists. The first shows the signatures beginning with N:

[19] For example, in John saw that Mary ate an apple, Suzy slept, and Lissy read a book. the extrapropositional coordination serves as the clausal object of *see*. Here the initial clausal conjunct *eat* would be &-marked.

7.6.1 Intrapropositional relations beginning with N

1 N/N ?
2 N/V subject/predicate
3 N/A ?

4 N\N ?
5 N\V object\predicate
6 N\A ?

7 N|N adnominal prepositional modifier
8 N|V adverbial prepositional modifier
9 N|A ?

10 N−N noun–noun coordination
11 N−V ?
12 N−A ?

Seven of the elementary signatures do not appear to be used in English (indicated by ?).[20]

Next consider the list of elementary signatures beginning with V:

7.6.2 Intrapropositional relations beginning with V

1 V/N ?
2 V/V infinitive/predicate (15.6.1, 1)
3 V/A ?

4 V\N ?
5 V\V infinitive\predicate (15.6.1, 2)
6 V\A ?

7 V|N progressive|noun, infinitive|noun (15.6.1, 3)
8 V|V ?
9 V|A ?

10 V−N ?
11 V−V verb–verb
12 V−A ?

Eight of the elementary signatures do not appear to be used in English.[21]

Finally consider the list of elementary signatures beginning with A:

[21] Chinese and Russian seem to use 1 (N/N) and 3 (N/A) as nominal and adjectival copulae .

7.6.3 Intrapropositional relations beginning with A

1	A/N	?
2	A/V	?
3	A/A	?
4	A\N	?
5	A\V	?
6	A\A	?
7	A|N	adj|noun
8	A|V	adj|verb
9	A|A	?
10	A−N	?
11	A−V	?
12	A−A	adj–adj coordination

Nine of the elementary signatures do not appear to be used in English.

In summary, of the 36 possible intrapropositional elementary signatures, English seems to use only 12. This may be summarized as follows:

7.6.4 Intrapropositional relations of English

1. N/V (subject/predicate)
2. N\V (object\predicate)
3. A|N (adj|noun)
4. A|V (adj|verb)
5. N−N (noun–noun)
6. V−V (verb–verb)
7. A−A (adj–adj)
8. V/V (infinitival_subject/predicate)
9. V\V (infinitival_object\predicate)
10. V|N (progressive|noun, infinitive|noun)
11. N|N (prepositional_phrase|noun)
12. N|V (prepositional_phrase|verb)

In addition, there are the extrapropositional relations (NLC Chaps. 7, 9). Of the 36 potential extrapropositional relations, English uses the following five:

7.6.5 EXTRAPROPOSITIONAL RELATIONS OF ENGLISH

13. V/V subject_clause/matrix_verb
14. V\V object_clause\matrix_verb
15. V|N adnominal_clause|noun, a.k.a. relative clause
16. V|V adverbial_clause|verb
17. V−V extrapropositional verb−verb

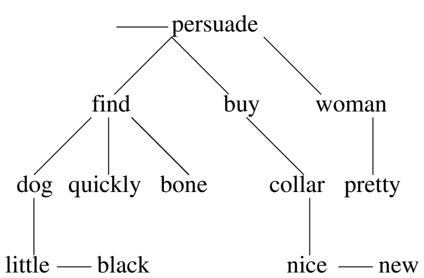
The relations 13, 14, 15, and 17 have intrapropositional counterparts, namely 8, 9, 10, and 6, respectively.

The small number of altogether 17 intra- and extrapropositional elementary signatures out of potentially 72 should make for easy language acquisition – at least as far as the composition of content is concerned.[22] The binary nature of elementary signatures provides the structural basis for the strictly time-linear, strictly surface compositional processing of natural language in DBS.

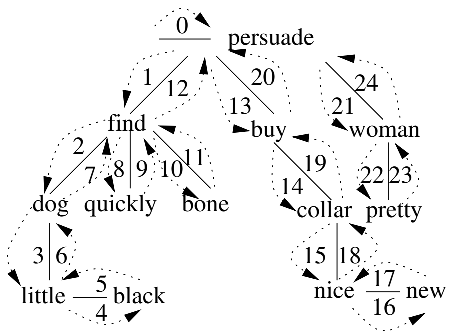
These basic principles hold for contents of arbitrary complexity. As an example consider the following DBS analysis of That the little black dog found the bone quickly persuaded the pretty young woman to buy a nice new collar:

7.6.6 CONTENT ANALYSIS CORRESPONDING TO A 20-WORD SENTENCE

(i) SRG (semantic relations graph)

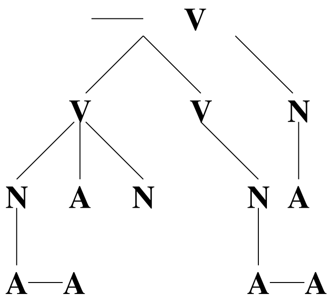


(iii) NAG (numbered arcs graph)



[22] Content is essential for language acquisition because it serves as the output of the hear mode and as the input to the speak mode.

(ii) signature



(iv) surface realization

1	2	3	4	5	6	7	10	11	8	9	12
That	the	little	black		dog	found	the_bone		quickly		persuaded
V/V	V/N	N|A	A−A	A−A	A|N	N/V	V\N	N\V	V|A	A|V	V/V

21	22	23	24	13	14	15	16	17	18	19	20
the	pretty	woman		to_buy	a	nice	new		collar		.
V\N	N|A	A|N	N\V	V\V	V\N	N|A	A−A	A−A	A|N	N\V	V\V

All subject/predicate, object\predicate, modifier|modified, and conjunct−conjunct relations are characterized explicitly in the (i) semantic relations graph and the (ii) signature. The (iii) NAG provides a depth first traversal order which is used for specifying the application sequence of the (iv) surface derivation. Even though the signature is semantic and has an initial node, it is not a semantic *hierarchy*.[23] Instead, the graph is a constellation of binary semantic relations of structure.[24]

The choice of which of two semantically related nodes must be the higher one in the graph is not determined by the core attributes, as witnessed by relations like V/V (7.5.2) or V|V (7.5.8). Instead, it is determined by the semantic relation of functor-argument between two nodes, which in turn constrains the possible choices of the core attributes, and accordingly for the horizontal order in coordination. This holds at the elementary, the phrasal, and the clausal level (7.1.3, 9.6.1–9.6.3, 12.6.3) of grammatical complexity.

[23] For example, there is no subclass or subtype relation between the V and the N in an N/V or any other elementary signature. For the treatment of subclass relations in DBS see Sect. 6.5.

[24] While every hierarchy may be represented as a graph, not every graph represents a hierarchy.

8. Building Nonlanguage Content

A problem for the time-linear derivation order of the DBS hear mode was *suspension*, i.e. the phenomenon that a next word proplet may not be connected immediately to any of the proplets in the current sentence start. For example, the next word Mary cannot be connected to the sentence start Yesterday in the derivation of Yesterday Mary danced (NLC Sect. 11.4). Instead, both have to wait for the arrival of danced, whereby the semantic relation connecting *yesterday dance* is A|V and the relation connecting *Mary dance* is N/V.

The problem of suspension was solved by (i) collecting the proplets of the sentence start and the proplet of the next word at the now front (word bank, 4.1.1) and (ii) applying hear mode operations whenever possible (self-organization). This solution of an empirical problem led to a simplification in the definition of DBS.Hear grammars because self-organization dispenses with the rule packages of LA grammar. Applying recognition operations whenever the input permits provides a method for handling a case in which no strict derivation order is provided externally by a sequence of language surfaces, namely nonlanguage recognition (convergence, FoCL Sect. 9.5).

The difference between nonlanguage and language recognition may be summarized as follows. First, the agent's nonlanguage recognition has to deal with a multi-dimensional environment, while language recognition deals with a one-dimensional language sign. Second, language interpretation is based on a strictly time-linear sequence of raw input, while no such order is provided for nonlanguage recognition. Third, natural language uses function and content words, while nonlanguage recognition deals with contents only.

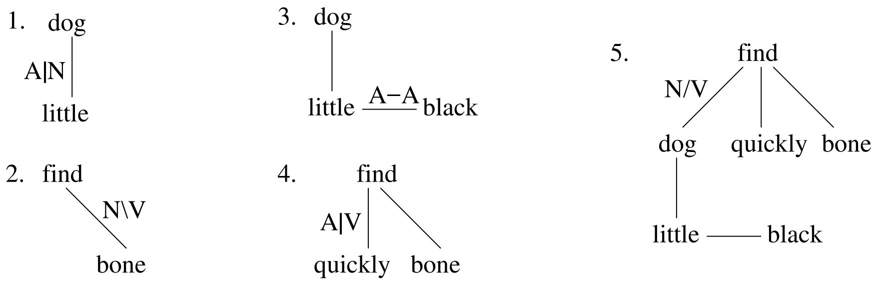
8.1 Intuitive Outline of DBS.Content

In DBS, analyzed word forms and elementary nonlanguage contents are treated alike as proplets. The only difference is a non-empty sur slot in language proplets and an empty sur slot in nonlanguage (context) proplets. In this way, the meaning of a word is treated formally as a content attached to a language-dependent surface by means of a convention (2.1.1, 3.1.1).

This similarity between the analysis of elementary language and nonlanguage items was originally motivated by treating reference as a pattern matching between the language and the context level (4.3.2, 4.3.3). The parallel treatment of language and nonlanguage cognition was continued further by storing proplets with the same core value in the same token line (6.4.3), regardless of their sur value. Placing incoming proplets at the now front in the token line with corresponding core and owner values puts them in their final storage position and limits the number of candidates for current composition.

The question of nonlanguage recognition is how nonlanguage contents should be combined without having an order imposed by agent-external language surfaces. As an example consider the derivation of the content corresponding to English The little black dog found the bone quickly.

8.1.1 Build-up of a nonlanguage content



The proplets are represented by their core value. The strictly local, binary relations between two individual proplets are intrapropositional signatures listed in 7.6.4. The hypothetical order of binary concatenations at the now front is indicated by the numbers 1 to 5, and is determined by the random arrival of the raw nonlanguage data assumed to be recognized and interpreted by the agent.

The process of establishing the relations might begin simultaneously with the A|N (*little|dog*) concatenation in 1 and the N\V (*bone\find*) concatenation in 2. Then these two separate content parts might each be extended simultaneously with the A−A (*little−black*) concatenation in 3 and the A|V (*quickly|find*) concatenation in 4, respectively. In 3, it is the first (lower) node of the A|N signature derived in 1 which is being extended. In 4 it is the second (upper) node of the N\V signature derived in 2 which is used for the extension. Finally, the top nodes of the two separate content parts 3 and 4 are concatenated in 5 with the N/V (*dog/find*) relation.

Any individual node, either isolated or already connected to other nodes, can be a local growth point in that it may be used as the first or second element of an elementary signature. For example, whether or not the nodes *dog* and *find* in concatenation 5 are already connected to other nodes (as shown in 3 and 4) does not matter as far as their local, binary concatenation is concerned. The main distinction in local concatenations (12.6.2) is between obligatory connections, i.e. subject/predicate and object\predicate,[1]and optional connections, i.e. modifier|modified and conjunct−conjunct.

The amalgamation of nonlanguage content is based on the following steps:

8.1.2 Steps of an elementary amalgamation

1. Raw data provided by the agent's visual, auditory, and other perception components are classified by concept types provided by the agent's memory, based on the principle of best match (Sect. 8.5; NLC Sect. 4.3; L&I'05) – as in a Rorschach test.
2. The instantiated concept tokens are embedded as core values into N, V, or A proplet shells (6.6.1, 6.6.5–6.6.7).
3. Selected pairs of nodes resulting from 2 are connected such that they form one of the 17 elementary signatures[2] defined in 7.6.4 and 7.6.5.

The pairs serving as input to the elementary signatures are unordered such that {N, V} and {V, N}, for example, are equivalent. There are six possible input pairs, namely {N, N}, {A, A}, {V, V} {A, N}, {A, V}, and {N, V}. Consider which node pairs are matched by which signature(s), using the numberings of 7.6.4 and 7.6.5:

8.1.3 Unambiguous and ambiguous input pairs to signatures

Unambiguous match for {A, N}
3. A|N (e.g. beautiful|dog)
Unambiguous match for {A, V}
4. A|V (e.g. beautifully|sing)
Unambiguous match for {A, A}
7. A−A (e.g. little−black)
Ambiguous match for {N, N}
5. N−N (e.g. man−woman)
8. N|N (e.g. in_Paris|house)

[1] The number and nature of the obligatory arguments are determined by the core value of the predicate.

[2] Language and thought are closely related by using the same core attributes for language and nonlanguage proplets, and connect language proplets with the same semantic relations of structure as nonlanguage proplets. This is part of the moderate linguistic relativism of DBS (Sect. 3.6).

Ambiguous match for {N, V}
1. N/V (e.g. John/gave)
2. N\V (e.g. Mary\gave)
9. N|V (e.g. in_Paris|live)
12. V|N (e.g. burning|fire, to_help|desire)
15. $\mathrm{V}|_x\mathrm{N}$ (e.g. who_loves|Mary)[3]

Ambiguous match for {V, V}
6. V−V (e.g. walk−talk)
10. V/V (e.g. to_err/is)
11. V\V (e.g. to_read\try)
13. $\mathrm{V}/_x\mathrm{V}$ (e.g. that_bark/surprise)
14. $\mathrm{V}\backslash_x\mathrm{V}$ (e.g. that_bark\hear)
15. $\mathrm{V}|_x\mathrm{V}$ (e.g. when_bark|smile)
17. $\mathrm{V}-_x\mathrm{V}$ (e.g. read−sleep)

The relations (3), (4), and (7) are unambiguous in English,[4] while the other constellations show a growing number of readings.

These ambiguities are possibilities of a formal nature. Whether they have counterparts in actual contents depends on the meaning of the parts to be related. Consider, for example, the choice between a coordination N−N (5) and a modification N|N (8): given the nouns pepper and salt, they may be combined as pepper and salt (N−N), but not as as N|N; conversely, the N in_Paris and the N house may be combined as house in_Paris (N|N), but not as N−N.

In summary, the 17 elementary signatures allow any N, V, or A node to be connected to any other N, V, or A nodes. In an unambiguous input pair, there is no need to choose between different signatures for relating the two nodes. For an ambiguous input pair, in contrast, the choice between the possible elementary signatures must be based on additional information, either (i) structural restrictions such as allowing only one subject or (ii) lexical properties of the concepts involved, such as the number and kind of valency positions.

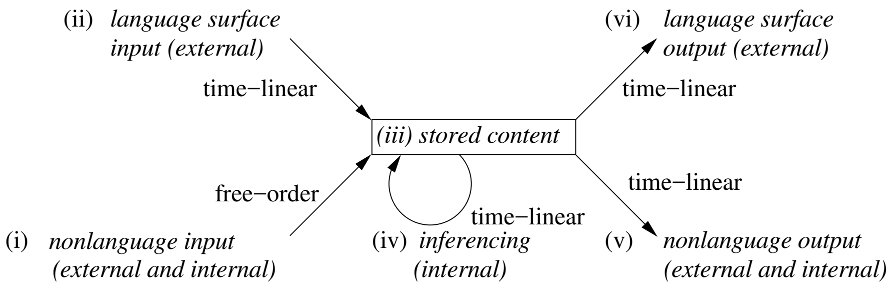
8.2 Formal Definition of DBS.Content

A cognitive agent with language has the following sources of content: (i) current nonlanguage recognition, (ii) current language interpretation, (iii) selective activation of nonlanguage and language content stored in memory, and (iv)

[3] The subscripted x indicates an extrapropositional relation.
[4] In Chinese and Russian, (3) {A, N} and (5, 8) {N, N} contents, like Julia (is) pretty and Julia (is a) doctor, may be analyzed as N/A and N/N, respectively, resulting in ambiguous matches.

inferencing which derives new content from current and stored content. These contents, regardless of their source, may be processed into blueprints which serve as input to the agent's action components, namely (v) nonlanguage action and (vi) language production (see also 14.3.1):

8.2.1 INTERACTION OF THE SIX SOURCES OF CONTENT



The derivation order of (ii) natural language interpretation in the hear mode, (iii) selective activation of content by navigating along the semantic relations between proplets in the think mode, (iv) inferencing on current and coactivated content, (v) nonlanguage action, and (vi) natural language production in the speak mode is strictly time-linear. For nonlanguage recognition (i), however, let us consider the possibility of free (random) order derivations.

The term "free order" is opposed to "fixed order" and applies to the steps of a recognition procedure, while the term "order-free" applies to the set of proplets representing a content.[5] In a free order derivation of nonlanguage content the composition follows the agent's attention wandering through the accidental highlights of an unfamiliar multidimensional environment.

Assuming a close similarity between language and nonlanguage content (Sects. 1.6, 3.6, 6.6), DBS.Content as defined in 8.2.2 below models the 17 elementary semantic relations of structure listed in 7.6.4 and 7.6.5 using free order derivations. The only available items are order-free proplets, ontologically representing objects (N), relations (V), and properties (A). The only semantic relations between items are functor-argument, represented by "/", "\", and "|" as well as coordination, represented by "−".

A DBS.Content grammar[6] is the non-language (context) counterpart of a DBS.Hear grammar. However, because there are no counterparts to function words in nonlanguage recognition, there is no function word absorption in the

[5] An order-free set of proplets is obtained by coding all interproplet relations proplet-internally as address values (3.2.8), thus allowing storage in any database schema of choice (here a content-addressable word bank) without loss of interproplet relations.

[6] The definition of DBS.Content benefitted from comments and suggestions by J. Handl.

operations of DBS.Content. Instead the recognition operations are all based on cross-copying, with two proplet patterns for matching input items and two proplet patterns for deriving an elementary signature (semantic relation of structure between the two proplets) as output, similar to 3.4.1.

In contradistinction to the operations of DBS. Nav. (3.4.2) and DBS.Speak (3.4.3), which traverse existing relations between proplets, DBS.Hear (3.4.2) and DBS.Content (8.2.2) establish new semantic relations between proplets. At the same time, the names of DBS.Nav and DBS.Content are alike because they are based on the same semantic relations of structure. However, because DBS.Content establishes semantic relations without any indication of direction, it differs from DBS.Nav in that its operation names have lines without arrow heads.

8.2.2 Definition of DBS.Content

intrapropositional relations

1. N/V:[7] $\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg: } \\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } \beta \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg: } \alpha \\ \text{prn: K}\end{bmatrix}$

   Comment: The operation N/V at the context level corresponds to the intrapropositional subject-predicate relation in English. The common prn value K indicates an intrapropositional relation. The operation cross-copies the core values matched by α and β into the arg slot of the verb and the fnc slot of the noun. The operation cannot reapply to its output because it requires the first arg slot of the input verb to be empty.

2. N\V: $\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg: X} \\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: } \beta \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg: X } \alpha \\ \text{prn: K}\end{bmatrix}$

   Comment: The operation N\V at the context level corresponds to the intrapropositional object-predicate relation in English. By assuming X $\neq$ NIL, the only difference of this operation from operation 1 (N/V) is the *noninitial* position of α in the arg slot of the verb, indicating the role of a grammatical object. Depending on the lexical valency structure of the verb's core value, more than one object may be added (usually at most two).

3. A|N: $\begin{bmatrix}\text{adj: } \alpha \\ \text{mdd: } \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: } \beta \\ \text{mdr: } \\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{adj: } \alpha \\ \text{mdd: } \beta \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: } \beta \\ \text{mdr: } \alpha \\ \text{prn: K}\end{bmatrix}$

   Comment: The operation A|N at the context level corresponds to the intrapropositional relation between an adnominal modifier and a noun in English. The operation cross-copies core values matched by α and β into the mdr slot of the noun and the mdd slot of the adj. Multiple adnominal modifiers as in the fuzzy little black dog are treated as an intrapropositional coordination (7.1.3, 9.4.5).

4. A|V: $\begin{bmatrix}\text{adj: } \alpha \\ \text{mdd: } \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{mdr: } \\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{adj: } \alpha \\ \text{mdd: } \beta \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{mdr: } \alpha \\ \text{prn: K}\end{bmatrix}$

   Comment: The operation A|V at the context level corresponds to the intrapropositional relation between an adverbial modifier and a verb in English. The operation cross-copies the core values matched by α and β into the mdd slot of the adj and the mdr slot of the verb. A verb may have several modifiers (NLC 15.4.2).

5. N−N: $\begin{bmatrix}\text{noun: }\alpha\\ \text{nc: }\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: }\beta\\ \text{pc: }\\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: }\alpha\\ \text{nc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: }\beta\\ \text{pc: }\alpha\\ \text{prn: K}\end{bmatrix}$

Comment: The operation N−N at the context level corresponds to the intrapropositional coordination of two noun conjuncts in English. The operation cross-copies the core values matched by α and β into the pc slot of the second noun and the nc slot of the first noun.

6. V−V: $\begin{bmatrix}\text{verb: }\alpha\\ \text{nc: }\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{pc: }\\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: }\alpha\\ \text{nc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{pc: }\alpha\\ \text{prn: K}\end{bmatrix}$

Comment: The operation V−V at the context level corresponds to the intrapropositional coordination of two verb conjuncts in English. The operation cross-copies the core values matched by α and β into the pc slot of the second verb and the nc slot of the first verb.

7. A−A: $\begin{bmatrix}\text{adj: }\alpha\\ \text{nc: }\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{adj: }\beta\\ \text{pc: }\\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{adj: }\alpha\\ \text{nc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{adj: }\beta\\ \text{pc: }\alpha\\ \text{prn: K}\end{bmatrix}$

Comment: The operation A−A at the context level corresponds to the intrapropositional coordination between two adnominal conjuncts in English. The operation cross-copies the core values matched by α and β into the pc slot of the second verb and the nc slot of the first verb.

8. N|N: $\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: }\beta\\ \text{mdr: }\\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: }\alpha\\ \text{mdd: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: }\beta\\ \text{mdr: }\alpha\\ \text{prn: K}\end{bmatrix}$

Comment: The operation N|N at the context level corresponds to the intrapropositional relation between a prepositional phrase and a modified noun in English. It is opaque because the fnc attribute of the first noun is changed into an mdd attribute in the output. The operation cross-copies the core values matched by α and β into the mdr slot of the second noun and the mdd slot of the first noun.

9. N|V: $\begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\alpha\\ \text{mdr: }\\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: }\beta\\ \text{mdd: }\alpha\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\alpha\\ \text{mdr: }\beta\\ \text{prn: K}\end{bmatrix}$

Comment: The operation N|V at the context level corresponds to the intrapropositional relation between a prepositional phrase and a modified verb in English. It is opaque because the fnc attribute of the first noun is changed into an mdd attribute in the output. The operation cross-copies the core values matched by α and β into the mdd slot of the noun and the mdr slot of the verb.

10. V/V: $\begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{arg: }\\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{arg: }\alpha\\ \text{prn: K}\end{bmatrix}$

Comment: The operation V/V at the context level corresponds to the intrapropositional relation between an infinitive serving as the subject and the matrix verb in English. The operation is opaque because a fnc attribute is added to the first pattern in the output. In this way, an infinitive serving as subject is connected to the matrix verb. The arg slot of the infinitive may take the value of an object and of the implicit subject (control).

11. V\V: $\begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{arg: X}\\ \text{prn: }\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{arg: X }\alpha\\ \text{prn: K}\end{bmatrix}$

Comment: The operation V\V at the context level corresponds to the intrapropositional relation between an infinitive serving as an object and the matrix verb in English. As X $\neq$ NIL is assumed, the only difference of this operation from operation 10 (V/V) is the *noninitial* position of α in the arg slot of the second verb, indicating the role of an object.

12. V|N: $\begin{bmatrix}\text{verb: } \alpha \\ \text{arg:} \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: } \beta \\ \text{mdr:} \\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: } \alpha \\ \text{mdd: } \beta \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: } \beta \\ \text{mdr: } \alpha \\ \text{prn: K}\end{bmatrix}$

Comment: The operation V|N at the context level corresponds to the intrapropositional adnominal modifier relation between a progressive verb form and a noun in English, as in the burning fire. The operation is opaque because the arg attribute of the first input pattern is changed to mdd in the output. The operation cross-copies the core values matched by α and β into the mdr slot of the noun and the mdd slot of the verb.

extrapropositional relations

13. V/$_x$V: $\begin{bmatrix}\text{verb: } \alpha \\ \text{arg:} \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg:} \\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: } \alpha \\ \text{arg:} \\ \text{fnc: } (\beta\ \text{K+1}) \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg: } (\alpha\ \text{K}) \\ \text{prn: K+1}\end{bmatrix}$

Comment: The operation V/$_x$V at the context level corresponds to the relation between a subject sentence and its higher verb in English (NLC Sect. 7.1). The relation is extrapropositional, as indicated by the different prn values of the output. Therefore, the values resulting from cross-copying must include the prn value of the source proposition. The operation is opaque because the first input pattern is assigned an additional fnc attribute in the output. The operation cross-copies the core values matched by α and β into the arg slot of the second verb and the fnc slot of the first verb.

14. V\\$_x$V: $\begin{bmatrix}\text{verb: } \alpha \\ \text{arg:} \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg:} \\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: } \alpha \\ \text{arg:} \\ \text{fnc: } (\beta\ \text{K+1}) \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{arg: X } (\alpha\ \text{K}) \\ \text{prn: K+1}\end{bmatrix}$

Comment: The extrapropositional operation V\\$_x$V at the context level corresponds to the relation between an object sentence and its higher verb in English (NLC Sect. 7.2). Assuming X $\neq$ NIL, the only difference of this operation from operation 13 (V/$_x$V) is the *noninitial* position of α in the arg slot of the second verb, indicating the role of an object.

15. V|$_x$N: $\begin{bmatrix}\text{verb: } \alpha \\ \text{arg:} \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: } \beta \\ \text{mdr:} \\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: } \alpha \\ \text{arg:} \\ \text{mdd: } (\beta\ \text{K+1}) \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: } \beta \\ \text{mdr: } (\alpha\ \text{K}) \\ \text{prn: K+1}\end{bmatrix}$

Comment: The extrapropositional operation V|$_x$N at the context level corresponds to the relation between a relative clause and its higher noun in English (NLC Sects. 7.3 and 7.4). The operation is opaque because the first output pattern is assigned an additional mdd attribute. The operation cross-copies the core values matched by α and β into the mdd slot of the verb and the mdr slot of the noun.

16. V|$_x$V: $\begin{bmatrix}\text{verb: } \alpha \\ \text{mdr:} \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{mdr: X} \\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: } \alpha \\ \text{mdr:} \\ \text{mdd: } (\beta\ \text{K+1}) \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{mdr: X } (\alpha\ \text{K}) \\ \text{prn: K+1}\end{bmatrix}$

Comment: The operation V|$_x$V at the context level corresponds to the relation between an adverbial clause and its higher verb in English (NLC Sect. 7.5). The operation is opaque because the first input pattern is assigned an additional mdd attribute in the output. The operation cross-copies the core values matched by α and β into the mdr slot of the second and the mdd slot of the first verb.

17. V$-_x$V: $\begin{bmatrix}\text{verb: } \alpha \\ \text{nc:} \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{pc:} \\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: } \alpha \\ \text{nc: } (\beta\ \text{K+1}) \\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta \\ \text{pc: } (\alpha\ \text{K}) \\ \text{prn: K+1}\end{bmatrix}$

Comment: The operation V$-_x$V at the context level corresponds to an extrapropositional coordination in English. It is transparent and cross-copies the core values matched by α and β into the pc slot of the second verb and the nc slot of the first verb, respectively.

Under normal circumstances (8.1.2), the agent's cognition will select (i) one concept interpretation for an elementary raw input, (ii) one proplet shell, and (iii) one elementary signature for relating two proplets resulting from (ii). However, as witnessed by the phenomenon of optical illusions, a given raw input may have more than one interpretation.

Applied to DBS.Content, this phenomenon makes it possible for a raw input to match (a) more than one concept, (b) more than one proplet shell, and/or (c) more than one elementary signature. Thereby, (a) and (b) correspond to a lexical ambiguity in natural language interpretation, and (c) corresponds to a syntactic-semantic ambiguity (8.1.3).

Recursive (systematic) ambiguities, however, may be avoided by "plugging the holes" with conditions some of which may be outside the grammar's operations. For example, the verbs in the lexicon have a maximum of three valency positions (e.g. [cat: n′ d′ a′ v]). Though not explicitly expressed in its abstract operation system, this is relevant for the complexity of DBS.Content.

8.3 The Linear Complexity of DBS.Content

It is well-known that the only source of complexity in the class of constant LA grammars (C-LAGs) is *recursive ambiguities* (TCS). The parsing of an input is recursively ambiguous if (i) it is recursive and (ii) the recursion repeats the same kind of ambiguity, resulting in a polynomial or exponential number of derivation steps (parallel readings).[8] Formal languages with recursive ambiguity are WW^R, accepting input like abcd...dcba, and WW, accepting input like abcd...abcd.... In these two formal languages the recursive ambiguity arises in worst-case inputs like aaaaaa.[9]

C-LAGs for input without any recursive ambiguity form a sub-class called C1-LAGs and parse in linear time. The C1-LAGs parse many context-sensitive languages, such as $\mathsf{a}^k\mathsf{b}^k\mathsf{c}^k$, $\mathsf{a}^k\mathsf{b}^k\mathsf{c}^k\mathsf{d}^k$, and so on, a^{2^i},a^{n^2}, $\mathsf{a}^{i!}$, etc.[10]

[8] See FoCL Sects. 11.3 and 11.4.

[9] In PSG, the complexity of WW^R is context-free (n^3), while that of WW is context-sensitive (2^n or worse; we know of no explicit phrase structure grammar for this simple language). In LA grammar, in contrast, both are C2-languages (n^2 polynomial). The formal LA grammars for WW^R and WW are defined in FoCL 11.5.4 and 11.5.6, respectively.

[10] A context-sensitive structure in natural language is The square roots of 16, 9 and 4 are 4, 3 and 2, respectively (Klepmann 2008). See also Shieber (1985); FoCL 9.2.2.

Given the long history of attempting to fit natural language into the complexity hierarchy of phrase structure grammar (FoCL Sects. 8.4, 8.5), we would like to know how natural language fits into the complexity hierarchy of LAG, which is orthogonal to that of PSG. In DBS, the empirical question of whether the natural languages are C1-languages and therefore parse in linear time is equivalent to the question of whether there exist constructions which are not only ambiguous, but recursively ambiguous (FoCL Sect. 12.5).

Structures which have been analyzed as recursive[11] in linguistics are relatively numerous. Consider the following examples:

8.3.1 Natural language structures analyzed as recursive

1. *Intrapropositional coordination*
   Examples: (i) The man, the woman, the child, ..., and the cat (noun coordination); cf. operation 5 (N−N) in 8.2.2. (ii) Peter bought, peeled, cooked, cut, spiced, served, ..., and ate the potatoes (verb coordination); cf. operation 6 (V−V). (iii) The fuzzy clever little black hungry ... dog (adnominal coordination); cf. operation 7 (A−A).
2. *Extrapropositional coordination*
   Example: Julia slept. Susanne sang. John read.; cf. operation 17 (V$-_x$V) in 8.2.2, 3.2.4 for a proplet representation, and NLC Sect. 9.2 for a DBS analysis.
3. *Iterated*[12] *object sentences*
   Example: John said that Bill believes that Mary suspects that Suzy knows that Lucy loves Tom; cf. operation 14 (V$\backslash_x$V) in 8.2.2 and 9.4.1 for a DBS analysis. Related are the constructions of *unbounded* or *long distance dependency*, such as Who did John say that Bill believes that Mary suspects that Suzy knows that Lucy loves?, which are analyzed in NLC 7.6.2–7.6.5. Iterated object sentences may also serve as a subject sentence, as in That Bill believes that Mary suspects that Suzy knows that Lucy loves Tom surprised John. as well as an object sentence, as in John assumes that Bill believes that Mary suspects that Suzy knows ...

[11] According to Chomsky, recursion is an essential property of natural language structure (Hauser, Chomsky, and Fitch 2002). This has been challenged by Everett (2005, 2013), based on the Pirahã language of the Amazon basin. Everett's analysis has been criticized by Nevins, Pesetsky, and Rodrigues (2009), who argue on the basis of Everett's data that Pirahã does have recursion.

[12] In computer science, every recursive function may be rewritten as an iteration, and vice versa. The choice between the two usually depends on the application and associated considerations of efficiency.

4. *Iterated relative clauses*
   Example: The man who loves the woman who feeds the child who has a cat is sleeping; cf. operation 15 (V$|_x$N) in 8.2.2 and 9.3.2 for a DBS analysis.

5. *Gapping constructions*
   Examples: (i) Bob ate an apple, walked the dog, read the paper, had a beer, called Mary, ..., and took a nap. (subject gapping); cf. operation 1 (N/V) in 8.2.2 and NLC Sect. 8.5. (ii) Bob ate an apple, Jim a pear, Bill a peach, Suzy some grapes, ..., and Tom a tomato. (predicate gapping); cf. operations 1 (N/V) and 2 (N\V) in 8.2.2, and NLC Sect. 8.5. (iii) Bob bought, Jim peeled, Bill sliced, Peter served, and Suzy ate the peach (object gapping); cf. operation 2 (N\V) in 8.2.2, and NLC Sect. 8.6. (iv) Bob ate the red, the green, and the blue berries. (noun gapping); cf. operation 3 (A|N) in 8.2.2 and NLC Sect. 8.6 for a DBS analysis.

6. *Iterated prepositional phrases*
   Example: Julia ate the apple on the table behind the tree in the garden ...; cf. operation 8 (N|N) in 8.2.2 and 7.2.5 for a partial DBS analysis; NLC Sects. 15.2, 15.3.

Of these constructions, only the last one could be construed as recursively ambiguous: each time a new prepositional phrase is added (recursion), there is a systematic ambiguity between (i) an adnominal and (ii) an adverbial interpretation. By (needlessly) multiplying out the semantic readings in the syntax, one may obtain exponential complexity. However, as shown in FoCL Sect. 12.5, this is simply a bad approach; there is the empirically adequate alternative of *semantic doubling*, which is of linear complexity.

In short, a C-LAG may (i) contain recursive structures and (ii) have ambiguities, but still be a C1-LAG as long as the ambiguities are not recursive. Whether or not an LA grammar contains recursive ambiguities may be determined in two ways: (i) investigate the grammar by following the parallel possible continuations through the input conditions of the operations or (ii) investigate the content phenomena which the grammar is designed to model.

If a recursive ambiguity is detected[13] in the operation system of a C-LAG

[13] Handl (2012) presents a classification algorithm which determines for any C-LAG if it is not recursively ambiguous. The algorithm is proven to run in r^2, where r is the number of rules in the grammar. Because the algorithm is based on the formal rule patterns and the rule packages of the LA grammar to be analyzed, it does not take into account other properties which may also be relevant for complexity. Therefore, an LA grammar classified as non-recursively ambiguous by the algorithm is guaranteed to run in linear time, but an LA grammar classified as recursively ambiguous may still be of lesser

for a natural language, it may or may not be empirically legitimate. If not, the recursive ambiguity must be eliminated from the grammar by tightening the input pattern of certain operations (debugging). The first case, in contrast, would constitute an interesting but unlikely linguistic discovery.

The contents recognized by DBS.content at the context level are of the same kind as those recognized by DBS.Hear at the language level. Therefore, the complexity and the concomitant generative power of DBS.Content is sufficient if it equals that of DBS.Hear, which is linear. There may be ambiguities in nonlanguage recognition, as in optical illusions, but no recursive ambiguities.

In short, because recursive ambiguities do not occur as an empirical phenomenon in natural language, the complexity of free order DBS.Content need not exceed linear. Nativism took the opposite direction: because discontinuous elements do occur as an empirical phenomenon, the complexity of constituent structure analysis (FoCL 8.5.4 f.) had to be raised from context-free phrase structure grammar (n^3) to transformational grammar (undecidable).

8.4 Grammatical Aspects of Content

The order-free nature of proplets underlying (i) the amalgamation of nonlanguage content and (ii) the storage of language and nonlanguage content in a word bank holds at all levels of content complexity, for example in a text:

8.4.1 COMPLEXITY LEVELS OF CONTENT IN TEXT

	text
	chapter
	section
	paragraph
	sentence
clausal	clause
phrasal	phrase
elementary	content word

The higher levels, i.e. sentence, paragraph, section, etc., are built by extrapropositional relations on top of the clausal level (Sect. 9.6).

This coding of content is "grammatical" in the following sense: it is based on (i) a small finite set of attributes, (ii) a small finite number of proplet kinds based on lists of attributes, (iii) a small finite set of nonnumerical

complexity due to properties which the algorithm cannot recognize, for example, lexical properties.

grammatical values, (iv) numbers serving as values for specifying linear order, identity, and nonidentity, and (v) a finite set of core values represented by place holders (6.6.8).

The relations between proplets are coded solely by means of address values. The relations may be established in the random order of simultaneous amalgamation (8.1.1, context level) or the time-linear derivation order of the hear (3.3.1) mode (language level).

From a sign-based point of view, however, the relations may also be established by following the order of the three levels of grammatical complexity, either upward or downward. The order of grammatical complexity in DBS is (i) phrasal, (ii) elementary, (iii) clausal (7.1.3, Sect. 9.6, 12.6.3),

The following example shows an upward establishment of semantic relations, following the three complexity levels with the analysis of a content represented by the English surface The little black dog barked. Mary smiled.[14]

8.4.2 Building semantic relations bottom-up

clausal relations (level 3)	adj: little& mdd: dog nc: black prn: 23	adj: black mdd: pc: little& prn: 23	noun: dog fnc: bark mdr: little& prn: 23	verb: bark arg: dog nc: (smile 24) prn: 23	noun: Mary fnc: smile prn: 24	verb: smile arg: Mary pc: (bark 23) prn: 24
elementary relations (level 2)	adj: little mdd: nc: prn:	adj: black mdd: pc: prn:	noun: dog fnc: bark mdr: prn: 23	verb: bark arg: dog nc: prn: 23	noun: Mary fnc: smile prn: 24	verb: smile arg: Mary pc: prn: 24
phrasal relations (level 1)	adj: little mdd: dog nc: black prn: 23	adj: black mdd: pc: little prn: 23	noun: dog fnc: bark mdr: little prn: 23	verb: bark arg: dog nc: prn: 23	noun: Mary fnc: smile prn: 24	verb: smile arg: Mary pc: prn: 24
lexical lookup (level 0)	adj: little mdd: nc: prn:	adj: black mdd: pc: prn:	noun: dog fnc: mdr: prn:	verb: bark arg: nc: prn:	noun: Mary fnc: prn:	verb: smile arg: pc: prn:
core value	*little*	*black*	*dog*	*bark*	*Mary*	*smile*

The items at the lowest level are the core values. Inserting them into appropriate proplet shells (lexical lookup) results in unconnected lexical items (level 0). At the phrasal level 1, the optional intrapropositional modifier|modified relations are established. At the elementary level 2, the obligatory intrapropositional subject/verb relations are added. The clausal level 3 shows the extrapropositional coordination of two sentential conjuncts.

The format of 8.4.2 is compatible with a sign-based constituent structure approach insofar as no explicit distinction is made between the speak and the

[14] The example shows an extrapropositional coordination, a phrasal noun with an intrapropositional adnominal coordination, and two obligatory functor-arguments, one with a phrasal, the other with an elementary subject.

hear mode. Moreover, the bottom-up derivation corresponds to that of C grammar (FoCL Sects. 7.4, 7.5). More specifically, the phrasal level 1 relations are established by the categorial canceling rules as

$\text{black}_{N/N} \bullet \text{dog}_{N} \Rightarrow \text{black dog}_{N}$

$\text{little}_{N/N} \bullet \text{black dog}_{N} \Rightarrow \text{little black dog}_{N}$,

the elementary level 2 relations as

$\text{bark}_{N/V} \bullet \text{little black dog}_{N} \Rightarrow \text{bark little black dog}_{V}$

$\text{smile}_{N/V} \bullet \text{Mary}_{N} \Rightarrow \text{smile Mary}_{V}$, and

the clausal level 3 relations as

$\text{little black dog bark}_{V/V} \bullet \text{Mary smile}_{V} \Rightarrow \text{little black dog bark Mary smile}_{V}$.

This bottom up derivation order is orthogonal to the time-linear derivation order of DBS.

Alternatively, the semantic relations may be established in opposite direction, beginning with the language surfaces rather than the core values:

8.4.3 Building semantic relations top-down

surfaces	the	little	black	dog	barked	.	Mary	smiled	.
lexical lookup (level 0)	noun: n_1 fnc: mdr: prn:	adj: little mdd: nc: prn:	adj: black mdd: pc: prn:	noun: dog fnc: mdr: prn:	verb: bark arg: nc: prn:	pnc: .	noun: Mary fnc: prn:	verb: smile arg: pc: prn:	pnc: .
clausal relations (level 3)	noun: n_1 fnc: mdr: prn:	adj: little mdd: nc: prn:	adj: black mdd: pc: prn:	noun: dog fnc: mdr: prn:	verb: bark arg: nc: (smile 24) prn: 23		noun: Mary fnc: prn:	verb: smile arg: pc: (bark 23) prn: 24	
elementary relations (level 2)	noun: n_1 fnc: mdr: prn:	adj: little& mdd: dog nc: black prn: 23	adj: black mdd: pc: little& prn: 23	noun: dog fnc: bark mdr: little& prn: 23	verb: bark arg: dog nc: (smile 24) prn: 23		noun: Mary fnc: smile prn: 24	verb: smile arg: Mary pc: (bark 23) prn: 24	
phrasal relations (level 1)		adj: little mdd: dog nc: black prn: 23	adj: black mdd: pc: little prn: 23	noun: dog fnc: mdr: little prn: 23	verb: bark arg: nc: (smile 24) prn: 23		noun: Mary fnc: prn:	verb: smile arg: pc: (bark 23) prn: 24	

After lexical lookup (level 0), the extrapropositional relations are established at level 3 with coordination and function word absorption of the periods. At level 2, the elementary relations of subject/predicate are established. At level 1, the phrasal relations are established with modifier|modified, conjunct−conjunct, and absorption of the determiner.

Like 8.4.2, the format of 8.4.3 does not distinguish between the speak and the hear mode and is therefore compatible with a sign-based approach. Moreover, the top-down derivation order corresponds to that of PS grammar (FoCL Sects. 8.1, 8.3, 8.4). More specifically, the clausal level 3 relations are established by the rewrite rule

S →S S,

the elementary level 2 relations by the rewrite rule

S → NP VP, and

the phrasal level 1 relations by the rewrite rule

NP → ADJ NP.

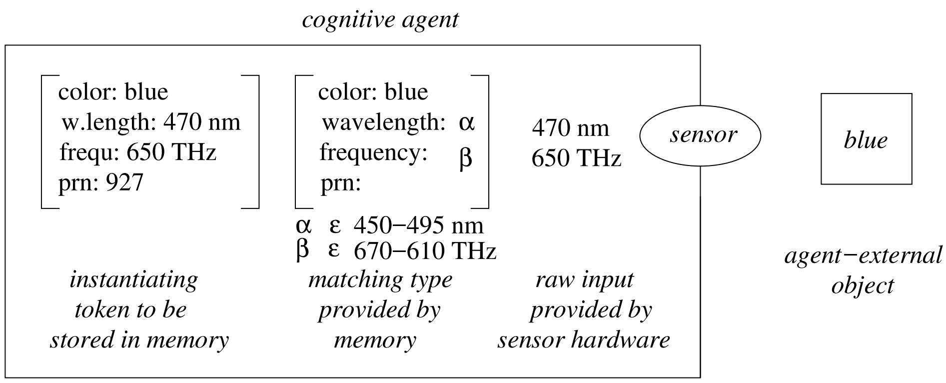
Despite their sign-based nature, the representations 8.4.2 and 8.4.3 differ from a corresponding constituent structure 12.2.1 in that DBS replaces (i) the dominance relation in a phrase structure tree by the three levels of phrasal, elementary, and clausal complexity and (ii) the precedence relation by order-free relations between proplets. Furthermore, while a phrase structure tree represents the semantic relations only indirectly (7.1.1, 7.1.2), they are coded directly in 8.4.2 and 8.4.3 by means of attributes and address values. These values define the levels 1–3 explicitly by their presence or absence; the diagonal lines are merely an optical aid to emphasize the relevant relations.

8.5 Defining Concepts Procedurally

DBS defines basic concepts as the elementary recognition and action procedures of an agent, and uses them as semantic primitives (grounding). Concepts have a declarative and a procedural specification which are based on the agent's (i) sensor hardware providing raw data and (ii) the agent's memory providing concept types for their classification (recognition) and realization (action). Concepts appear as one of the three sign kind of natural language, the others being the indexicals and the names.

The procedural definition of concepts constitutes the bottom level of the DBS robot's interaction with external reality. As an example, consider an artificial agent recognizing an instance of the color *blue*:

8.5.1 CONCEPT TYPE AND CONCEPT TOKEN IN RECOGNIZING *blue*



The agent's color sensor measures the electro-magnetic wavelength 470 nm and the frequency 650 THz in an agent-external object. These values lie within the intervals 450–495 nm and 670–610 THz, which constitute the type of the color blue and are provided by the agent's memory. In the resulting token, the wavelength and frequency intervals of the type are replaced by the measured values and a prn value is provided.

Next consider the type and the token of a geometric object:

8.5.2 Type and token of the concept *square*

type (FoCL 3.3.1)

$$\begin{bmatrix} \text{edge 1: } \alpha \text{ cm} \\ \text{angle 1/2: } 90^{\circ} \\ \text{edge 2: } \alpha \text{ cm} \\ \text{angle 2/3: } 90^{\circ} \\ \text{edge 3: } \alpha \text{ cm} \\ \text{angle 3/4: } 90^{\circ} \\ \text{edge 4: } \alpha \text{ cm} \\ \text{angle 4/1: } 90^{\circ} \end{bmatrix}$$

where α is a length

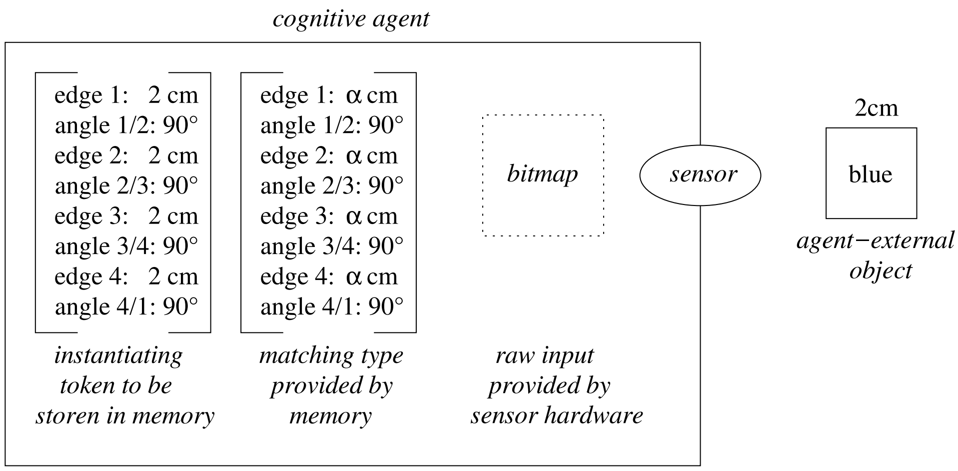
token (FoCL 3.3.2)

$$\begin{bmatrix} \text{edge 1: 2 cm} \\ \text{angle 1/2: } 90^{\circ} \\ \text{edge 2: 2 cm} \\ \text{angle 2/3: } 90^{\circ} \\ \text{edge 3: 2 cm} \\ \text{angle 3/4: } 90^{\circ} \\ \text{edge 4: 2 cm} \\ \text{angle 4/1: } 90^{\circ} \end{bmatrix}$$

Here, the type and the token share attributes which specify (i) the number of equally long edges and (ii) the angle of their intersections. The type and the token differ only in their edge lengths. Specified in the type as the variable α, the edge length in the token is 2cm. The latter is accidental in that the type matches an infinite number of square tokens with different edge lengths. In an artificial agent, the type may be implemented as a pattern-matching software which recognizes tokens by approximating raw bitmap outlines (FoCL 3.2.1).[15]

The recognition of a square may be shown as follows:

8.5.3 Concept type and concept token in recognizing *square*



The type matches the outline of all kinds of different squares, whereby its variables are instantiated in the resulting tokens.

By embedding the types and tokens of elementary recognition and action as core values into proplets, the storage and retrieval mechanism of the agent's word bank (4.1.1) may be used. The core value of the proplet type serves as the owner value, while the proplet tokens are stored at the now front of the associated token line and will be left behind as member proplets when the now front is being cleared (Sect. 13.3). Associating the hardware of particular sensors and actuators with certain token lines provides an efficient method for finding the token line of a token for storage and retrieval.

In DBS, the procedural method for establishing core values is used also for the interpretation and synthesis of language surfaces (NLC Sect. 4.5). For example, the English surface blue is recognized at the language level by applying a matching type to the raw data and instantiating it as a token, just as the color *blue* is recognized at the context level, and similarly for action. In talking agents, nonlanguage and language cognition cooperate in that nonlanguage contents with core value types are attached to the surface of content words, serving as their meaning$_1$.

8.6 Memory-Based Pattern Completion

As hardware devices, the number of sensors and actuators in an agent is finite and manageable. Though the amount of raw data perceived and generated by these devices is limited only by the agent's lifetime, the number of types provided by memory for classifying or realizing them is finite and determined by the agent's survival tasks in its ecological niche. Complexity arises in the composition of elementary data into complex contents (8.1.1).

Regarding the recognition of complex contents, cognitive psychology distinguishes two basic approaches. The *atomistic* approach tries to compose complex contents from a few basic elements, called features,[16] used many times. The *holistic* approach tries to provide templates for each relevant constellation of basic elements, composition is best not used at all.

[15] Natural counterparts to abstract elementary recognition in terms of concept types and tokens are the line, edge, and angle detectors in the visual cortex of the cat, discovered by Hubel and Wiesel (1962).

[16] Computer science defines the term feature as an attribute-value pair (avp), such that feature structures consist of features. Cognitive psychology, in contrast, uses feature for the basic elements of recognition and action. Yet another use is proposed in ISO 24610-1: to reserve the term attribute for some XML convention, an attribute-value pair (avp) is renamed a "feature-value pair." This is unfortunate because feature = feature value has the form of an inappropriate diagonalization.

Computationally, these alternatives have inverse, complementary merits. For example, if there are only 50 different kinds of objects to be recognized, a holistic approach using templates would be reasonable: the cost of running through 50 templates in the worst case is compensated by the efficiency of the matching. However, if there are a thousand different kinds of objects to each be matched as a whole, the cost of checking all templates in the worst case would outweigh any advantage in matching (if there is any left, given the increased complexity[17] of the templates).

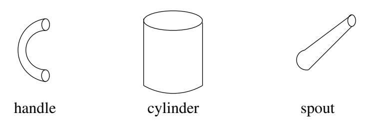
An approach making the most of the inverse merits of the holistic and the atomistic approach is the RBC (Recognition-by-Components) or geon theory by Biederman (1987 et seq.). It analyzes visual recognition in terms of neither basic features nor holistic templates, but rather in terms of some intermediate structures, called geons, from which all the parts for complex objects are built.

RBC is described by Kirkpatrick (2001) as follows:

> The major contribution of RBC is the proposal that the visual system extracts geons (or geometric ions) and uses them to identify objects. Geons are simple volumes such as cubes, spheres, cylinders, and wedges. RBC proposes that representations of objects are stored in the brain as structural descriptions. A structural description contains a specification of the object's geons and their interrelations (e.g. the cube is above the cylinder).
> . . .
> The RBC view of object recognition is analogous to speech perception. A small set of phonemes are combined using organizational operations to produce millions of different words. In RBC, the geons serve as phonemes and the spatial interrelations serve as organizational rules. Biederman (1987) estimated that as few as 36 geons could produce millions of unique objects.

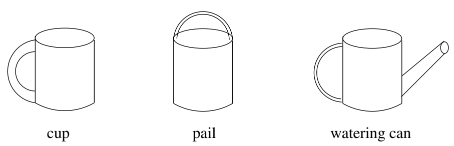
Consider the following examples of geons:

8.6.1 A SMALL SET OF ISOLATED GEONS



By matching these geons onto corresponding parts of the visual input, different complex objects may be analyzed as being composed of a small set of connected geons:

8.6.2 ANALYZING DIFFERENT OBJECTS AS CONNECTED GEONS

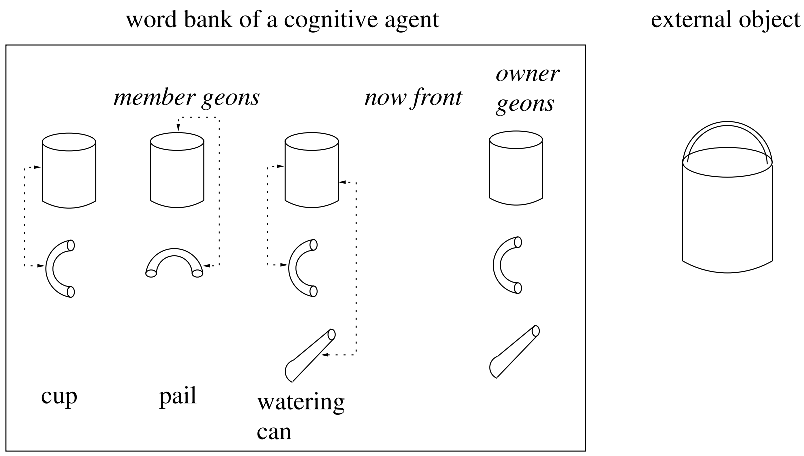


As a theory of cognitive psychology, geons claim the advantage of being *viewpoint invariant*: based on laboratory experiments, Biederman and Gerhardstein (1993) have argued[18] that subjects can distinguish and recognize geons from different perspectives. Also, the theory is economical in that 24 geons and a small set of relations such as on-top-of, larger-than, end-to-end, end-to-middle, and so on, can be combined into 306 billion possible 3-geon combinations.

Computationally, however, such a large number of possible combinations presents the task of defining an efficient algorithm for determining the correct relations between the geons found in a visual input. For this, DBS uses the distinction between (i) sensor data activating cognition and (ii) cognition controlling sensors. For example, when a medical doctor sees a rash (sensor data activating cognition), (s)he may actively look for some tell-tale visual cue provided by memory to determine the particular kind (cognition controlling sensors), assuming that (s)he is a well-trained and experienced expert.

These complementary procedures are supported by the database schema of a word bank. As an example, consider the storage of the connected geons 8.6.2 in a word bank:

8.6.3 Storing connected geons in a word bank



In this word bank, the isolated geons are the owners and the connected geons are the members. For intuitive appeal (and as a substitute for an explicit formal definition of the inter-geon relations), the connections are indicated by

[17] The increase of complexity follows from the necessity to differentiate the large number of individual templates from each other.

[18] For critical comments see Tarr and Bülthoff (1995).

dotted bi-directional arrows. The complex objects specified in each column of connected geons are provided with names, i.e. cup, pail, and watering can.

If the cylinder is recognized first (sensor controlling cognition), the word bank will indicate that cylinders are known to be connected in certain ways to handles and/or spouts. This information is used to actively analyze the cylinder's relations to the rest of the external object (cognition controlling sensors), checking for the presence or absence of the items suggested by the data base in the raw input (memory-based pattern completion, L&I'05).

Similarly, if the handle is recognized first, the agent's database indicates that handles are known to be connected in certain ways to cylinders to form cups, pails, or watering cans. If the handle-cylinder connection is recognized first, there are two possibilities: pail or watering can. In our example, the system determines that there is no spout and recognizes the object as a pail.

This simple algorithm may be described somewhat more generally as follows: Assume that the agent is faced with a complex object consisting of an open number of geons and that one of these, e.g. B, has been recognized. Then the database will retrieve all B items and list all their connections to other geons. The geons on this list are all tried actively on the external object by checking (i) whether they are present or absent in the external object and, if present, (ii) the nature of their connection to B.

As soon as a second fitting geon, lets call it C, has been found, the algorithm will retrieve all BC connections of the given type from the database, as well as all their connections to other geons. This results in a substantially smaller list. By repeating the process, the algorithm converges very quickly. Efficiency may be improved even further by taking frequency and domain information into account (Sect. 15.5).

9. Aspects of Language Content

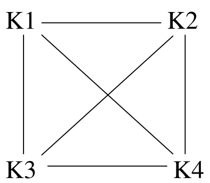
This chapter investigates relations between relatively language-independent contents and the associated language-dependent surfaces. Section 9.1 presents DBS graphs for different linguistic constructions which have the same degree sequence. Section 9.2 analyzes alternative word orders within a language as different navigations through the same NAG. Section 9.3 shows the same method for word order differences between languages. Section 9.4 compares clausal contents which consist of the same propositions, but use different relations to connect them. Section 9.5 reconstructs the linguistic notions of marked/unmarked in DBS. Section 9.6 discusses the degrees of elementary, phrasal, and clausal grammatical complexity.

9.1 Graph Theory

The binary nature of semantic relations in DBS makes them ideal for a graph-theoretical analysis: elementary signatures like N/V, N\V, or A|N may be represented uniformly as two uninterpreted nodes (vertices) connected by one uninterpreted line (edge). In this way, the ontologically motivated distinctions between different core attributes (nodes) and different kinds of semantic relations (lines) may be abstracted away from. What is focused on instead is the branching structure and the accessibility of nodes.

Central notions of graph theory are the *degree* of a node and the *degree sequence* of a graph. Consider, for example, the n=4 (four-node) graph known as K_4.

9.1.1 THE COMPLETE N=4 GRAPH K_4

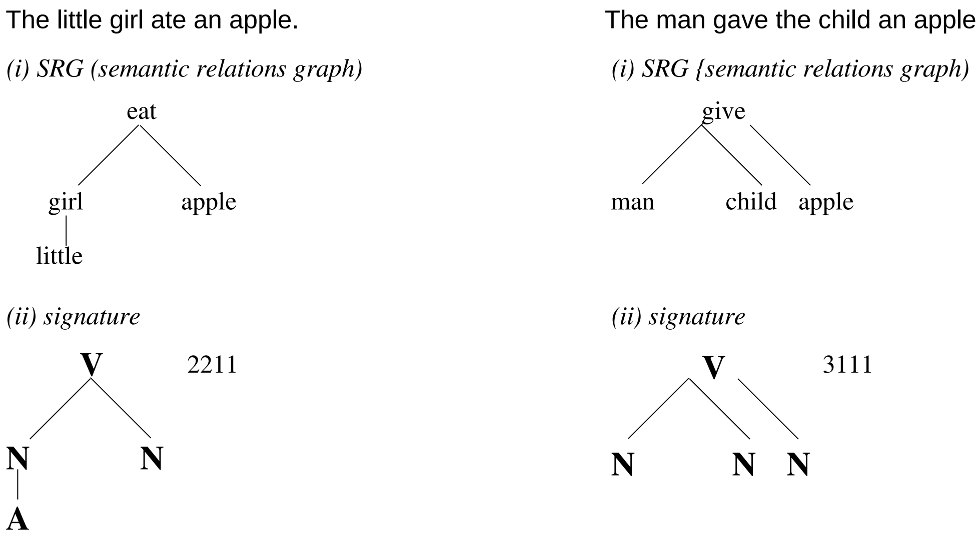


This graph is called *complete* because each node is connected to all the other nodes. The degree sequence is 3333 because each of the four nodes is of degree 3 (i.e. connects three lines).

The order of digits in a degree sequence begins with the highest degree, followed by the next highest degree, all the way down to the lowest degree nodes in the graph. Accordingly, the degree sequence[1] of the signature in 7.4.1 is 2211, in 7.6.6 is 4322222211111, and in 9.2.5 is 211. In contrast to K_4, these linguistically motivated signatures are not complete.

The following DBS graphs on the left and the right have the same number of nodes – and therefore the same number of digits in their respective degree sequences, but corresponding digits show different degrees:

9.1.2 Same number of nodes but different degrees



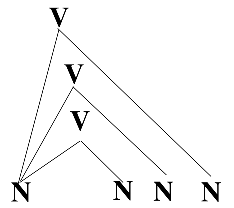
The graphs on the left have a two-place verb with its arguments and the subject is modified by an adjective, while the graphs on the right have a three-place verb with its arguments and no modification. Accordingly, the respective degree sequences have the same number of digits, namely four, but the digits have different degrees, namely 2211 on the left and 3111 on the right.

It is also possible for different constructions to have identical degree sequences. As an example consider the signatures of a subject gapping (NLC 8.5.3) and a similar object gapping (NLC 8.6.3):

[1] The extrapropositional start line is disregarded.

9.1.3 Same number of nodes and same degrees

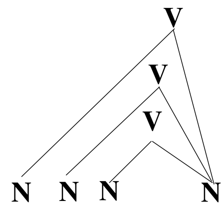
(ii) signature of a subject gapping



Bob bought an apple, peeled a pear and ate a peach.

degree sequence: 3222111

(ii) signature of a object gapping



Bob bought, Jim peeled, and Bill ate the apple.

This example shows that degree sequences alone can not provide all the distinctions needed for a complete linguistic analysis.

Nevertheless, graph-theoretical concepts allow us to formulate the following constraints on well-formed DBS graphs:

9.1.4 Graph-theoretical constraints on DBS graphs

1. The signature must be *simple*, i.e. there may be no loops or multiple lines.
2. The NAG must be *symmetric*, i.e. for every arc connecting some nodes A and B,[2] there must also be an arc from B to A (with the exception of extrapropositional coordination, Sect. 9.4).[3]
3. The traversal of arcs must be *continuous*, i.e. combining the traversal of arc x from A to B and of arc y from C to D is permitted only if B = C.
4. The numbering of arcs must be *exhaustive*, i.e. there must exist a navigation which traverses each arc.

Because of conditions 1 (simple) and 2 (symmetric), the (i) SRG, the (ii) signature, and the associated (iii) NAG are graph-theoretically equivalent.

In addition, a NAG must satisfy the following linguistic requirements:

9.1.5 Linguistic conditions to be satisfied by a NAG

1. An intrapropositional traversal must begin and end with the node which acts as the predicate (verb).[4]
2. The verbal top node must be entered either via the start line (arc 0) or by a corresponding arc from a preceding proposition (9.4.3).

---

[2] By requiring two nodes, a special exclusion of start lines is unnecessary.

[3] See NLC Chaps. 11 (hear mode) and 12 (speak mode) for a detailed analysis of extrapropositional coordination as the DBS complement of propositional calculus in symbolic logic. The combination of an intra- and an extrapropositional verb coordination is shown in NLC 9.1.2.

[4] Partial propositions without a verb, e.g. answers, certain headlines, etc., require special attention.

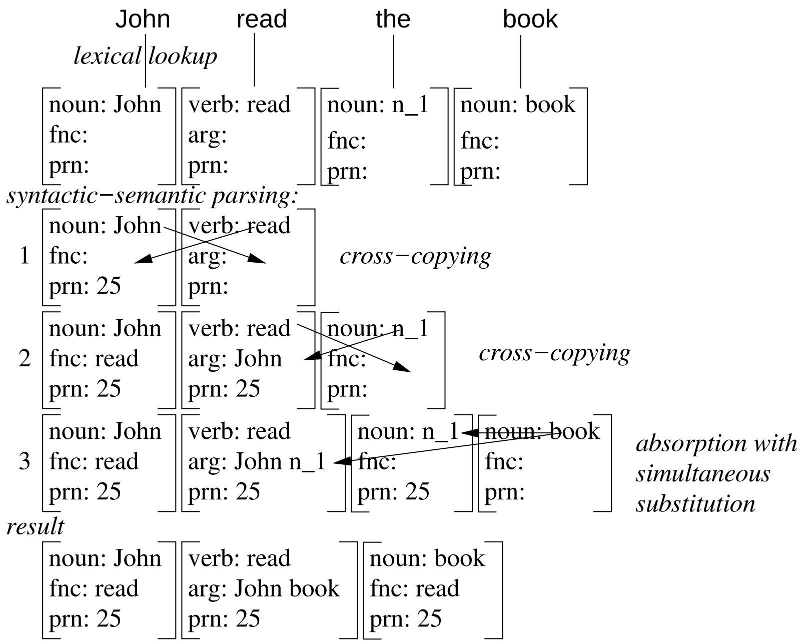
3. The only lines permitted in a NAG are "/" (subject/predicate), "\" (object\-predicate), "|" (modifier|modified), and "−" (conjunct−conjunct).
4. The only nodes permitted in a NAG based on a signature are N (noun), V (verb), and A (adjective).

Conditions 1 and 2 are necessary for the extrapropositional coordination of two propositions, as in a text.

9.2 Alternative Word Orders Within a Language

While the intrapropositional numbering of the arcs in a NAG is fixed to either depth first or breadth first (TExer Chap. 1) there may be alternative traversals. As an example, consider verbal voice in English,[5] i.e. the alternation between active and passive (Twiggs 2005). Let us begin with the hear mode derivations of John read the book (active) and The book was read by John (corresponding passive):

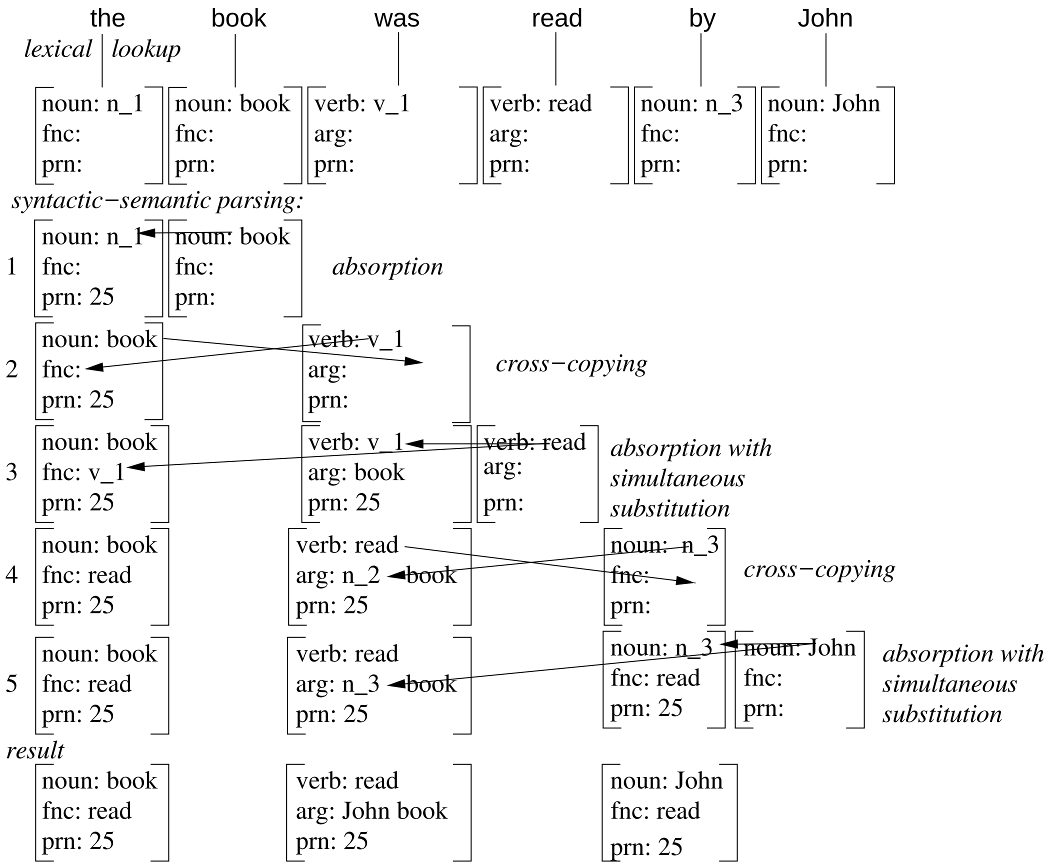
9.2.1 INTERPRETATION OF ACTIVE: John read a book



Anticipating the content 9.2.3 common to the active and the passive, let us use the prn value 25 also in the following derivation of the corresponding passive:

[5] There are other possible verbal voices, such as the *medium* (middle voice) in classical Greek.

9.2.2 INTERPRETATION OF PASSIVE: The book was read by John



Up to line 2, the voice of the verb is undecided: the sentence start the book was could be continued as an active, as in the book was lying on the table, or as the passive in question. However, as soon as the past participle of *read* is added in line 3, the order of the values in the arg attribute of the verb proplet is adjusted by inserting the nominal substitution value n_2, fixing the switch to passive. The result is shown by the verb proplet in line 4.

At this point, the sentence could be completed as The book was read. Such a suppression of the grammatical agent is an option typical of passive (Givon 1997). In line 4, it is expressed by the feature [arg: n_2 book] of *read*. It would be up to inferencing to determine the agent by finding a value for n_2. However, as the sentence continues with the *by*-phrase (7.2.1, 1b), the substitution value n_2 is replaced by n_3. The noun proplet *John* is added and absorbed into the *by*-proplet, replacing all occurrences of n_3 with John.

The result is the same content[6] as derived in the corresponding active. Consider the presentation as a set of proplets, using the alphabetical order induced by the core values, with added cat and sem features:

9.2.3 CONTENT COMMON TO ACTIVE AND PASSIVE

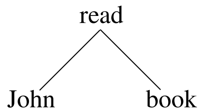
noun: book	noun: John	verb: read
cat: snp	cat: snp	cat: #n′ #a′ decl
sem: indef sg	sem: nm m	sem: past
fnc: read	fnc: read	arg: John book
mdr:	mdr:	mdr:
prn: 25	prn: 25	prn: 25

In this content, the subject/predicate relation is coded as N/V and the object\predicate relation as N\V.

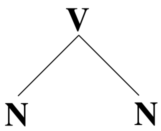
In the speak mode, active and passive are based on alternative navigations through the same NAG:

9.2.4 ALTERNATION BETWEEN ACTIVE AND PASSIVE IN ENGLISH

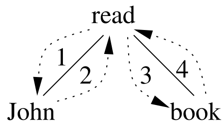
(i) SRG (semantic relations graph)



(ii) signature



(iii) NAG (numbered arcs graph)



(iv) surface realization

a. 1 2 3 4
John read a_book •
V/N N/V V\N N\V

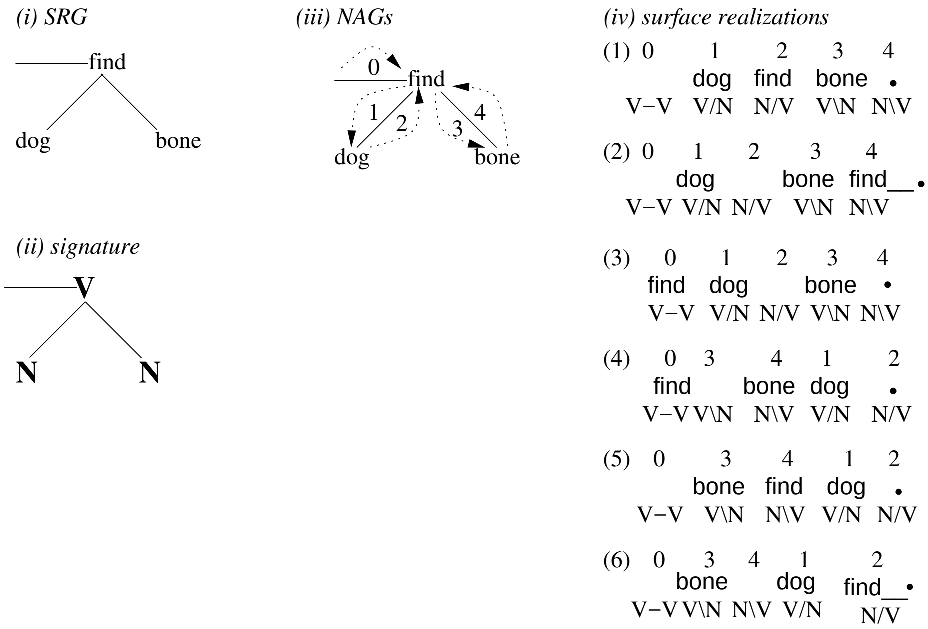
b. 3 4 1 2
A_book was_read by_John •
V\N N\V V/N N/V

The difference between the active and the passive variants is reflected by the alternative numbering of the two surface realizations. The a- and b-variants are an instance of *paraphrase* (FoCL 4.4.5). As in the alternation between The man gave the child an apple and The man gave an apple to the child (NLC 6.1.4), the variants show more than one surface for the same content and are based on alternative traversals of a given NAG.

[6] Whether an expression in the active voice and its passive counterpart have the same content or not has been discussed controversially in the literature. Originally generated from the same "deep structure" by a transformation, the meaning equivalence was later cast into doubt based on differences in quantifier scope. For example, Everyone in this room speaks at least two languages (active) and At least two languages are spoken by everyone in this room (passive) seem to correspond to the

Another use of alternative traversals through the same NAG is the computational reconstruction of free word order. For example, the free word order of Russian allows the following six possible permutations of {dog, find, bone} for coding the content *dog find bone*:

9.2.5 Russian word order based on alternative traversals



There are (i) one SRG, (ii) one signature, (iii) one NAG, and (iv) six surface realizations (1)–(6). The NAG begins with arc 0, traversing the start line. Given that surfaces may be realized only from the goal proplet of a navigation step,

---

alternative orders of quantifiers postulated by truth-conditional semantics (NLC 6.4.4). The elimination of quantifiers in DBS allows us to treat the literal meaning$_1$ of active and passive as equivalent, moving the alleged scope ambiguity into the inferencing of pragmatics, where it belongs.

Differences in the pragmatic interpretation of order extend from the active-passive paraphrase to coordination: while p∧q and q∧p are equivalent in symbolic logic (rule of symmetry), they have different implications in language, as shown by following example by G. Lakoff (lecture at the linguistic summer school UC Santa Cruz, 1971 or 1972):

John opened the window and threw out the cat.
John threw out the cat and opened the window.

As another example consider similarity. From a logical point of view, one may assume that the similarity relation is symmetric, such that If A resembles B, then B resembles A. Yet the following example shows that alternative orders may express differences in appraisal.

This Ford resembles a Mercedes.
This Mercedes resembles a Ford.

arc 0 is not only motivated by extrapropositional coordination (9.4.3), but is also needed for verb-first surfaces, such as the Russian versions of find dog bone and find bone dog.

Realizations (1) and (2) share a 0–1 traversal to realize dog. (1) continues with an arc 2 traversal to produce find, an arc 3 traversal to produce bone, and an arc 4 traversal to produce the period. (2) uses traversals 2–3 to produce bone and 4 to produce find_.

Realizations (3) and (4) share traversal 0 to realize find. (3) continues with traversal 1 to produce dog, traversals 2–3 to produce bone, and 4 to produce the period. Realization (4) continues with traversal 3 to produce bone, traversals 4–1 to produce dog, and 2 to produce the period.

Realizations (5) and (6) share a 0–3 traversal to realize bone. (5) continues with traversal 4 to produce find, traversal 1 to produce dog, and traversal 2 to produce the period. (6) uses a traversal 3 to produce bone and 4–1 to produce dog, and 2 to produce find_.

9.3 Alternative Word Orders Across Languages

Alternative traversals also provide the basis for reconstructing word order differences in corresponding constructions of different natural languages.[7] For example, multiple relative clauses in German may be center-embedded, as shown by the following intuitive structural representation:

9.3.1 Relative clause center embedding

German:	Der Mann					singt
		der die Frau			liebt	
			die das Kind	füttert		
English transliteration:						
	the man	who the woman	who the child	feeds	loves	sings

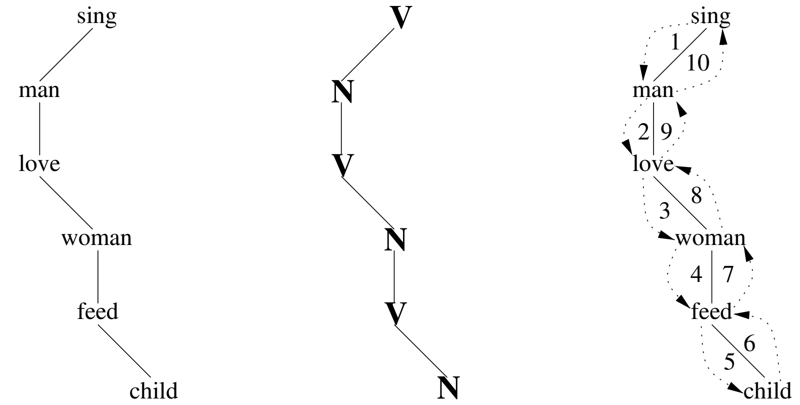
Based on the interpretation of the "/", "\", and "|" lines, the DBS graph analysis of this construction[8] turns out to be simple:

[7] If a construction in two natural languages or language classes uses different valency structures (lexicalization, 3.6.1), their SRGs, signatures, and NAGs may differ as well.

[8] For a general discussion of clausal adnominal modifiers, aka relative clauses, see NLC Sects. 7.3 (subject gap) and 7.4 (object gap).

9.3.2 GRAPH ANALYSIS OF CENTER-EMBEDDED RELATIVE CLAUSES

(i) SRG (semantic relations graph) *(ii) signature* *(iii) NAG (numbered arcs graph)*



(iv) surface realization (German, center−embedded)

1	2	3	4	5	6	7	8	9	10
Der_Mann	der	die_Frau	die	das_Kind	fuettert		liebt		singt_.
V/N	N|V	V\N	N|V	V\N	N\V	V|N	N\V	V|N	N\V

The graph consists of six nodes (vertices) connected by five lines (edges) and has the degree sequence 222211. The signature is is composed from the elementary signatures N/V (subject/predicate), N\V (object\predicate), and V|N (clausal_modifier|noun). Because the semantic relations coded by these signatures are universal, speakers of different languages can agree where their DBS graph structures of this constructions are alike and where they differ,[9] for example, because of alternative lexicalization or alternative syntactic-semantic coding such as the *ergative* in Tagalog.

The German surface realization walks down on the left side of the NAG to realize the nouns, then back up on the right side to realize the verbs. Multiple realizations and multiple visits (9.3.6, 9.3.7) are evenly spread. Each arc is visited once and the traversal begins and ends with the main verb.

In comparison, consider the English counterpart:

9.3.3 ENGLISH REALIZATION OF CONTENT 9.3.2

The man who loves the woman who feeds the child sings.

Because English subclauses have the verb in post-nominative position (in contrast to the verb-final position in German subclauses, FoCL Sect. 18.3), the

[9] Thanks to the native speakers of many different languages (3.5.2) who participated in these enlightening discussions at the CLUE.

first relative clause may be completed before the second begins. Therefore almost all nouns and verbs may be realized on the way down on the left side of the NAG in 9.3.2:

9.3.4 English surface realization of relative clauses

surface realization (English, unmarked)

```
   1        2          3          4          5     6   7   8   9   10
The_man  who_loves  the_woman  who_feeds  the_child                sings_.
  V/N      N|V        V\N        N|V        V\N   N\V V|N N\V V|N  N/V
```

Going back up on the right side of the NAG, there is nothing left to do except at the very end (arc 10), when the main verb and the period are realized.

Another possibility of English word order is the *extraposition* of a relative clause, as in the following variant of 9.3.4:

9.3.5 Surface realization with extraposed relative clause

surface realization (English, extraposed, marked)

```
   1      10   1     2          3          4          5      6   7   8   9   10
The_man  sings     who_loves  the_woman  who_feeds  the_child                  .
  V/N     N/V V/N    N|V        V\N        N|V        V\N   N\V V|N N\V V|N N\V
```

The realization of this surface requires a multiple visit. After realizing the_man in arc 1, the navigation returns to the verb and realizes sings in arc 10. From there, there is no choice but to traverse arc 1 again and continue with arc 2 to the relative clause. Then the navigation travels down on the left side of the NAG in 9.3.2, realizing the verbs and the nouns. On the way back up, there is nothing left to do except to realize the period.

The surface realization 9.3.5 shows that a consecutive numbering is a sufficient, but not a necessary, condition for satisfying continuity (9.1.4, 3). For example, the combined traversal of arcs 1 and 10 is continuous even though the numbers are not consecutive.

From the software side, the mechanism of multiple visits is easily programmed. From the cognitive and linguistic sides, however, multiple visits must be constrained because otherwise there would be no limit on complexity. The constraint applies to connected graphs and comes in two variants.

9.3.6 Multiple visit constraint I (without language)

In content navigation without language realization, a multiple visit is

1. permitted if there are still untraversed arcs in the graph,
2. prohibited if all arcs in the graph have been traversed.

These conditions are based on the possibility of keeping track of how often a node has been traversed. Constraint I does not exclude multiple visits, but their number must be limited on the basis of traversal counters (FoCL 24.4.3).

9.3.7 MULTIPLE VISIT CONSTRAINT II (WITH LANGUAGE)

In content navigation with language realization, a multiple visit is

1. permitted if a required function word has not yet been realized,
2. prohibited if there is no value remaining in the current proplet set which has not been used exhaustively for realization.

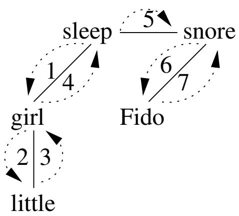
Technically, constraint II is based on language-dependent DBS.Speak grammars for mapping content into surfaces (NLC Chaps. 12, 14).

9.4 Handling Semantically Related Contents

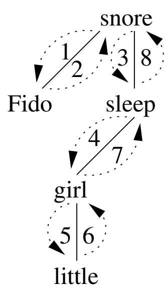
Apart from alternative surfaces for one and the same content discussed in the two preceding sections, there are contents which are semantically related, but structurally different. For example, The little girl slept. Fido snored. (extrapropositional conjunct−conjunct, parataxis) and When the little girl slept, Fido snored (extrapropositional modifier|modified, hypotaxis) consist of the same two propositions, but the connection between them is different. This may be illustrated by the following NAGs and associated surface realizations:

9.4.1 DIFFERENT NAGS FOR SEMANTICALLY RELATED CONTENTS

extrapropositional coordination



extrapropositional functor−argument



surface realization

1	2	3	4	5	6	7
The	little	girl	slept_.		Fido	snored_.
V/N	N|A	A|N	N/V	V−V	V/N	N/V

surface realization

3	4	5	6	7	8	1	2
When	the	little	girl	slept		Fido	snored_.
V|V	V/N	N|A	A|N	N/V	V|V	V/N	N/V

The two contents may be related by an inference like the following:

9.4.2 INFERENCE RELATING THE CONTENTS IN 9.4.1

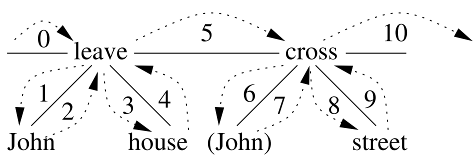
$$\begin{bmatrix}\text{verb: }\alpha\\ \text{nc: }\beta\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{verb: }\beta\\ \text{prn: K+1}\end{bmatrix} \Longleftrightarrow \begin{bmatrix}\text{verb: }\beta\\ \text{mdd: }(\alpha\text{ K})\\ \text{prn: K+1}\end{bmatrix}\begin{bmatrix}\text{verb: }\alpha\\ \text{mdr: }\beta\\ \text{prn: K}\end{bmatrix}$$

There are two extrapropositional relations, V−V on the left and V|V on the right. If an input matches the left pattern, the construction on the right is derived,[10] and vice versa.

Another example of two contents being related but having different NAGs may be constructed with alternative codings of a temporal sequence, as in John left the house. Then he crossed the street. and John crossed the street. Earlier he left the house. Syntactically, this constitutes two propositions A and B coordinated either as AB or as B′A′.

Compared to intrapropositional coordination, extrapropositional coordination in the speak mode is special in that it is simplex (16.6.7, iii). Consider the following NAGs:

9.4.3 EXTRAPROPOSITIONAL COORDINATION (SIMPLEX)



1	2	3	4	5	6	7	8	9
John	left	the_house	•	Then	he	crossed	the_street	•
V/N	N/V	V\N	N\V	V−V	V/N	N/V	V\N	N\V

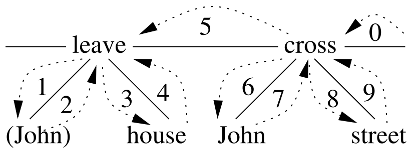
In text- or dialog-initial position, the initial "−" coordination line 0 without a preceding node serves as the start line. The subsequent extrapropositional "−" lines 5 and 10 have a left-hand node, representing the verb of the preceding proposition. Arcs 0, 5, and 10 are simplex because for production in the speak mode there is no need for a return from, e.g., *cross* to *leave*.

Inverting the order of the two propositions with a concomitant recoding of the temporal order requires a different semantic structure:

---

[10] The choice of an appropriate subordinating conjunction is the task of conditions which are left aside.

9.4.4 Coding the content of 9.4.3 with inverted order



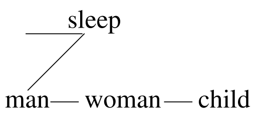
6	7	8	9	5	1	2	3	4
John	crossed	the_street	.	Earlier	he	left	the_house	.
V/N	N/V	V\N	N\V	V−V	V/N	N/V	V\N	N\V

Compared to 9.4.3, arc 5 is inverted. The alternative surface order and different temporal conjunctions are shown in the surface realization. The relation between word order and the traversal of the NAG is specified by the arc numbering in the top line of the surface realization. The temporal order (9.4.3) and the anti-temporal order (9.4.4) are related by an inference like 9.4.2.

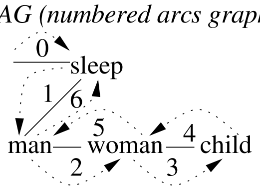
In contrast to extrapropositional coordination, intrapropositional coordination must be duplex to enable the return of the navigation to the top verb, needed for the continuation to a next proposition. Consider the following example:

9.4.5 Intrapropositional noun coordination (duplex)

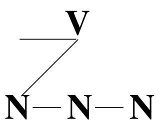
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	3	4	5	6
The_man	the_woman	and_the_child			slept_.
V/N	N−N	N−N	N−N	N−N	N\V

The duplex coding of the two intrapropositional coordinations consists in the arcs 2, 5, and 3,4. The traversal of arcs 4 and 5 is empty.

9.5 Marked vs. Unmarked

Alternative traversals (paraphrases) of a NAG may be related to the distinction between unmarked and marked (Givón 1997). For example, the consecutive numbering of *active* (9.2.1) may be regarded as unmarked, while the nonconsecutive numbering of the *passive* (9.2.2) as marked. Similarly, the indirect object without a preposition directly following the verb, e.g. The man gave the child an apple, has a consecutive numbering and may be regarded as unmarked and the variant with prepositional object following the direct object, e.g.The man gave an apple to the child, as marked (NLC 6.1.4) In each of these cases, the unmarked and marked paraphrase are based on the same transparent content.[11]

Before turning to alternative orders, let us list the ordering principles which hold for every navigation. First, traversal of an extrapropositional conjunct (9.4.3) must start from the highest predicate as the unique entry point.[12] Second, to get to an argument, a navigation must first traverse the functor. Third, to get to a modifier, a navigation must first traverse the modified.[13] Fourth, to get to a non-initial conjunct, a navigation must first traverse the initial conjunct. Fifth, once the end of a branch has been reached, the navigation must return to the beginning of the branch to continue with the next branch.

For example, in 9.4.5 the extrapropositional navigation entering the proposition (arc 0) encounters the matrix verb *sleep* first. From this entry point it proceeds to the initial conjunct of the subject coordination *man* (arc 1), continues to the second (arc 2) and third (arc 3) conjunct, returns empty to the first conjunct (arcs 4, 5), and navigates back to the initial entry point (arc 6) to realize the verb. If the top verb's nc (next conjunct) attribute has a value, the navigation may continue to a next proposition (9.4.3).

The possibility of alternative orders arises when a navigation is faced with more than one available branch. This is shown by the comparison between the active (9.2.4, a) and the corresponding passive construction (9.2.4, b). The alternative branches are the semantic relations N/V (subject/predicate) and N\V (object\predicate) at the top of the graph. Active traverses the NAG branches top down and from left to right. Passive traverses them top down and from right to left. These alternatives use the same NAG, but result in the different surface realizations (paraphrase).

The arc numbers in the top line of a surface realization may be seen as being purely descriptive. Once they have been recorded they may be used to enforce a certain navigation order through the NAG for realizing a given content, represented by the SRG, in a certain language.[14] The other possibility is to let the navigation find the path through the graph on its own.

[11] See 9.2.3 for the content common to an active and a corresponding passive.

[12] A special case is subject and object gapping (TExer 1.4.5–1.4.7).

[13] This is why the levels of grammatical complexity are ordered (i) phrasal, (ii) elementary, (iii) clausal in a DBS graph (Sect. 9.6).

This may be based on two complementary principles. The first, called the *downward principle*, is the condition that all nodes should be traversed during a navigation (9.1.4, 4); it ensures that a new branch is entered if one is available. The second, called the *upward principle*, is the condition that an intrapropositional navigation must end where it began (9.1.4, 2); it ensures that the navigation through a branch returns to the beginning. Giving the downward principle slightly higher weight enables it to apply again until all downward nodes have been traversed. Then the upward principle guides the navigation back to the point of entry, i.e. the top verb.

By relating the marked/unmarked distinction to the principle of least effort,[15] it may be integrated into the autonomous navigation: (i) traversing a branch completely obeys least effort while exiting early and then requiring a multiple visit (NLC 7.6.2; TExer 5.5.17) does not; (ii) following the time-linear order inherent in an intrapropositional construction the content obeys least effort while changing to another order does not (9.4.4); (iii) continuing the direction of an extrapropositional navigation obeys least effort, while reversing the direction does not (9.2.4).

The reason why passive and similar marked traversals are used in communication anyway is due to other principles which override least effort. For example, theme-rheme (topic-comment), WH question formation, and so on, induce word order requirements of their own which are often in marked contrast to the order of what would be least effort.

The marked/unmarked distinction does not apply if there is no syntactic alternation for the same content. For example, taken in isolation the word order of Julia put the flowers in a vase (NLC 6.4.4) might be called marked, but there is no variant for the obligatory prepositional object. Conversely, the word order of the adnominal reading in Julia ate the apple on the table (NLC 15.2.1) might be called unmarked taken by itself, but there is no alternation for this content in the unanalyzed (FoCL 4.4.4) surface.

9.6 Levels of Grammatical Complexity

In addition to the three core attributes noun, verb, and adj, there are the five grammatical roles of *subject, object, predicate, modifier*, and *conjunct*, and the three levels of grammatical complexity. The latter may be ordered (i) "naturally," i.e. *elementary, phrasal, clausal* (9.6.1; NLC 15.1.1), or (ii) according

[14] With the possibility of taking frequency into account.

[15] See T. Givón (1990). Optimality theory uses markedness for the ranking of constraints (Kager 1999).

to the position of proplets in a DBS graph, i.e. *phrasal, elementary, clausal* (7.1.3, 8.4.2, 8.4.3, 9.6.2).

The grammatical roles and the levels of complexity are orthogonal to each other. Consider the following examples, using the natural order of complexity:

9.6.1 Grammatical roles and complexity levels

	elementary	*phrasal*	*clausal*
subject	books; she; Julia	the beautiful girl	that Fido barked
predicate	barked	could have barked	Fido barked.
modifier	here; beautiful	in the garden	when Fido barked

Accordingly, we say "phrasal noun" when we mean something like the man in the brown coat rather than "noun phrase."[16]

Proplets with different core attributes may be used for the same grammatical role. For example, (i) elementary beautifully with the core attribute adj, (ii) phrasal on the table with the core attribute noun[17] (7.2.5), and (iii) clausal when Fido barked with the core attribute verb may take the same grammatical role, namely adverbial modifier.

What they have in common is the semantic relation of structure concatenating the modifier and the modified. In the DBS graph, the relation is represented by "|", as in A|V (elementary), N|V (phrasal), and V|V (clausal). In the representation of content as a set of proplets, the modifier|modified relation is coded by the mdd value of the modifier and the mdr value of the modified. In elementary adjectives, the mdd attribute originates in the lexical entry (NLC A.6.1), in prepositional phrases in the lexical entry of the preposition (NLC A.4.5), and in clausal modifiers in the lexical entry of the subordinating conjunction (NLC A.4.5). In the modified, here the V proplet (NLC Sect. A.5), the mdr attribute is standard.

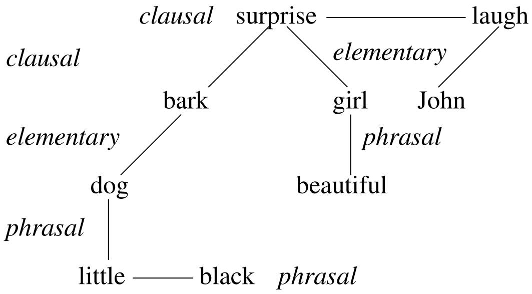
To see why an elementary relation is positioned *between* a clausal and a phrasal relation, if present, in a DBS graph.[18] let us consider the SRG for That the little black dog barked surprised the beautiful girl. John laughed., i.e. a subject subclause construction coordinated with a simple sentence:

[16] Except in the discussion of the Bach-Peter sentence (11.4.9) in which we implicitly adopt the terminology of transformational grammar.

[17] In prepositional phrases, the attributes noun and mdd are introduced by the lexical analysis of the preposition (NLC A.4.5). In classical Latin, nouns are used as modifiers in the ablative case, which may express location, instrument, or time. English uses nouns syntactically as modifiers in expressions like kitchen|table (two words), which other languages handle in the morphology as compounds, for example, Küche/n/tisch with a *Fugen-n* in German.

[18] I.e. an elementary relation is positioned below a clausal and above a phrasal relation.

9.6.2 EXTRAPROPOSITIONAL FUNCTOR-ARGUMENT CONSTRUCTION

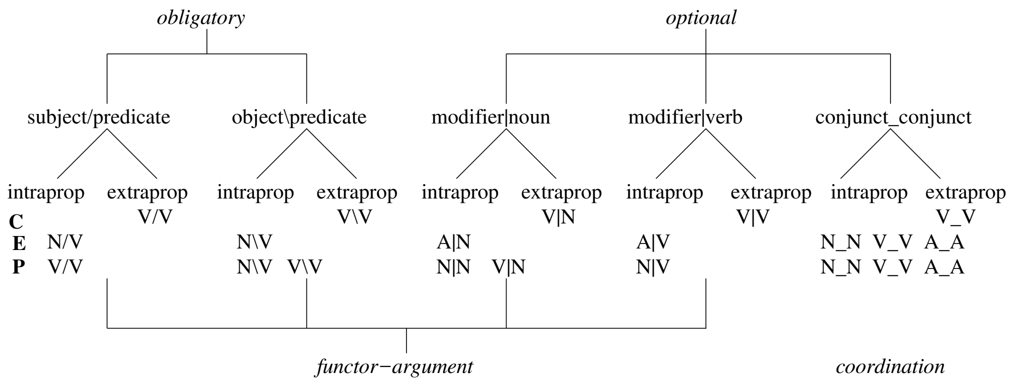


The extrapropositional coordination between the top verbs *surprise* and *laugh* is clausal. The relations to their arguments may be clausal, as between *surprise* and *bark*, or elementary, as between *bark* and *dog*, *surprise* and *girl*, and *laugh* and *John*. The elementary arguments may be modified by the phrasal relation between *dog* and *little*, and between *girl* and *beautiful*. Initial modifiers may be extended into an intrapropositional coordination, as in *little black*, which is phrasal because it is part of a phrasal noun.

The relations of a certain complexity need not all appear at the same level in the graph. For example, the relations *dog*/*bark* and *John*/*laugh* are both elementary, but appear at different levels.

The relations between (i) the five semantic relations of structure, (ii) the three core attributes N, V, and A, and (iii) the three levels of phrasal (P), elementary (E), and clausal (C) grammatical complexity are realized in 17 elementary signatures. Listed in 7.6.4 (intrapropositional), and 7.6.5 (extrapropositional), they may be summarized schematically as follows:

9.6.3 RELATING SEMANTIC RELATIONS TO LEVELS OF COMPLEXITY



Phrasal V/V and V\V stand for infinitival subjects and objects. The C(lausal) row corresponds to 7.6.5, the E(lementary) row to the relations in 7.6.4 (1-7), and the P(hrasal) row to the relations in 7.6.4 (8-12).

Because there are no nonterminal nodes in a DBS content, each semantic relation of structure is defined directly between two content nodes, regardless of the level of grammatical complexity. A node, however, may have more than one semantic relation to other nodes. For example, in 9.6.2 the node girl is in the optional phrasal relation beautiful|girl and at the same time in the obligatory elementary relation girl\surprise.

Furthermore, the node surprise is in the extrapropositional (clausal) coordination relation surprise/laugh, the obligatory clausal subject/predicate relation bark/sleep, and in the obligatory elementary relation surprise\girl. In other words, the *surprise* node has degree 3 (Sect. 9.1). Altogether, the graph 9.6.2 has the degree sequence 322222111.

Extrapropositional (clausal) coordination is exceptional in that only one direction is used for a content navigation (simplex) – in contradistinction to all the other semantic relations of structure, which must be traversed in both directions in order to return to the top level (duplex). Compared to functor-argument, neither intra- nor extrapropositional coordination has an obligatory realization (i.e. is always optional).

Coordination at all levels in English and many other languages is also special in that conjuncts must have the same core attribute. If this is true for all natural languages, the number of theoretically possible relations of structure listed in 7.6.1, 7.6.2, and 7.6.3 may be reduced by 14[19] from 72 to 58.[20]

[19] Six for intrapropositional and eight for extrapropositional coordination.

[20] Technically, the putative universal is easily realized in DBS because a core value may be embedded into proplet shells of different core attributes (6.6.5– 6.6.7). For example, if desired, the coordination in Mary enjoys swimming, riding, books, and music. may be made to satisfy the coordination universal by embedding the "verbal" core values swim and ride into nominal proplet shells.

10. Computing Perspective in Dialog

Agent-based DBS is designed for robots which are capable of spontaneous dialog with humans in a natural language of choice. This requires not only the automatic processing of language expressions, but of *utterances*. DBS defines an utterance as (i) a language content connected to (ii) a value cluster called STAR (Sects. 11.1). The STAR is continually updated by the agent's onboard orientation system (Sect. 14.2) and serves to anchor a content to the parameter values of Space, Time, Agent, and Recipient. At the same time, a STAR provides referents for the *indexicals*, which are one of the three sign kinds.

For transporting a content from the speaker to the hearer, the basic data of the respective onboard orientation systems are used to compute three different perspectives in real time, represented by three different STARs. The STAR-0 records current recognition and action (monitoring) and does not involve language. The STAR-1 is the speaker's perspective on content used for language production. The STAR-2 is the hearer's perspective on content resulting from language interpretation. Proceding from the agent's current STAR-0, computing the STAR-1 and the STAR-2 is by means of inferences and completely software-mechanical.

10.1 Agent's STAR-0 Perspective on Current Content

Sign-theoretically, indexicals raise the question of whether they are function or content words. In DBS, all words with their own reference mechanism, i.e. the symbol, the indexical, and the name (HBTR Sects. 3.2–3.4), are treated as content words (*autosemantika*[1]). All words which do not have their own reference mechanism, such as determiners, prepositions, conjunctions, and auxiliaries, are treated as function words (*synsemantika*).

The reference mechanism of indexicals consists in pointing at current STAR parameter values. The agent's onboard orientation system provides five parameters: for (1) the speaker, as in I and we, (2) the hearer, as in you, (3) an

[1] Marty (1908), pp. 205 f.

agent or object other than the speaker and the hearer,[2] as in he, she, it, they, and this (FoCL Sects. 17.2, 18.2), (4) the present moment, as in now, and (5) the present location, as in here (Sect. 14.2). In language communication, the referent of an indexical is determined by the *origin* of the sign containing it, represented as the STAR-1 of the sign's utterance (FoCL Sect. 5.3).

Before a content can be mapped into natural language surfaces (speak mode) it must emerge as nonlanguage content in the agent's cognition. Nonlanguage content monitoring of the agent's current situation is anchored to the STAR-0:

10.1.1 Anchored nonlanguage content

I am writing you a letter.$^{\mathrm{STAR-0}}$

This informal representation specifies the content in two parts:[3] (i) a sentence of English, used here as a means to represent a nonlanguage content, and (ii) a STAR-0, added as a superscript at the end, called the anchor.[4] The content represented by the English sentence alone is called unanchored. In DBS, this part of 10.1.1 is coded as the following set of proplets:

10.1.2 Coding unanchored content as a proplet set

noun: pro1	verb: write	noun: pro2	noun: letter
fnc: write	arg: pro1 pro2 letter	fnc: write	fnc: write
prn: 659	prn: 659	prn: 659	prn: 659

The values of the associated STAR-0 may be specified as follows:

10.1.3 Specification of a STAR

S = Paris
T = 1930-07-03
A = Jean-Paul Sartre
R = Simone de Beauvoir

[2] This parameter is not reflected in STAR as an acronym, but represented in explicit STARs by the attribute 3rd when needed (10.1.6).

[3] This notation is reminiscent of Montague's (1974, p. 258) use of "indices." However, while the @,i,j,g index cluster in Montague's PTQ refers to a set-theoretically defined model structure @, a possible world i ϵ I, a moment of time j ϵ J, and a variable assignment g (Sect. 13.2), the STAR refers to the agent-external real world. For the DBS treatment of Space and Time see Sect. 14.2.

[4] The distinction between a content and its anchor may be applied to any system recording its current state. Systems may differ, however, with respect to the attributes and the range of values of their anchor. In avionics and air traffic control, for example, the anchor would use attributes like ground location, altitude, speed, and direction (*situation awareness*, Endsley et al. 2000, 2003) rather than space, time, author, and recipient.

The value of the A(gent) attribute is self-referential and constant for the agent. The value of the T(ime) attribute is continuously incremented. The values of the S(pace) and R(ecipient) attributes may also change. Therefore, a content must be connected to the agent's current STAR-0 as soon as it emerges.

The connection between an unanchored content and its STAR-0 is formally established by defining the STAR-0 as a proplet with the same prn value as the associated propositional content. The following example complements the unanchored proplet representation 10.1.2 with the STAR-0 anchor 10.1.3:

10.1.4 STAR-0 CONTENT WITH 1ST AND 2ND PERSON INDEXICALS

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: write}\\ \text{prn: 659}\end{bmatrix} \begin{bmatrix}\text{verb: write}\\ \text{arg: pro1 pro2 letter}\\ \text{prn: 659}\end{bmatrix} \begin{bmatrix}\text{noun: pro2}\\ \text{fnc: write}\\ \text{prn: 659}\end{bmatrix} \begin{bmatrix}\text{noun: letter}\\ \text{fnc: write}\\ \text{prn: 659}\end{bmatrix} \begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-03}\\ \text{A: J.-P. Sartre}\\ \text{R: S. de Beauvoir}\\ \text{prn: 659}\end{bmatrix}$$

This nonlanguage content constitutes an agent-based perspective insofar as J.-P. Sartre is looking out towards his current location S, his current moment of time T, himself as an agent A in the world, and S. de Beauvoir as his addressee R (intended recipient, partner of discourse). As indexicals, pro1 and pro2[5] are defined to point at the A and R values, respectively, of the STAR-0.

An agent may also record an observation without any self-reference A or involvement of a recipient R, as in the following example.

10.1.5 STAR-0 CONTENT WITHOUT INDEXICALS

$$\begin{bmatrix}\text{noun: Fido}\\ \text{fnc: bark}\\ \text{prn: 572}\end{bmatrix} \begin{bmatrix}\text{verb: bark}\\ \text{arg: Fido}\\ \text{prn: 572}\end{bmatrix} \begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-03}\\ \text{A: S. de Beauvoir}\\ \text{prn: 572}\end{bmatrix}$$

In this anchored nonlanguage content, Madame de Beauvoir notes that Fido is barking. Because the content contains no indexical pointing at the R attribute, this feature may be omitted in the STAR-0 specification.

Alternatively to referring to Fido by name, Madame de Beauvoir could have used the 3rd person indexal pro3,[6] as in the following content corresponding to He barked.:

---

[5] In real life, Monsieur Sartre and Madame de Beauvoir addressed each other with vous. This relates to the register of honorifics, which is highly grammaticalized in Korean and Japanese, for example. As a socially important aspect of perspective, the register of honorifics (politeness) must be integrated into the R feature of the STAR.

[6] In the FoCL definition of the STAR, the attribute 3rd providing the referent for the third person indexical pro3 was not included because it does not participate directly in anchoring a sign's content. It is necessary, however, for the interpretation of the indexical (i.e. non-coreferential) uses of he, she, it, etc. (Chap. 11). Rather than changing the terminology from STAR to STAR3rd, we continue with the simpler term STAR, but use the attribute 3rd and the value pro3 for indexical uses of third person pronouns when needed (e.g. 10.1.6).

10.1.6 STAR-0 CONTENT WITH A 3RD PERSON INDEXICAL

$$\begin{bmatrix}\text{noun: pro3}\\ \text{fnc: bark}\\ \text{prn: 572}\end{bmatrix} \begin{bmatrix}\text{verb: bark}\\ \text{arg: pro3}\\ \text{prn: 572}\end{bmatrix} \begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-03}\\ \text{A: S. de Beauvoir}\\ \text{3rd: Fido}\\ \text{prn: 572}\end{bmatrix}$$

The indexical use of pro3 is defined to point at the value of the 3rd attribute of the STAR-0. In contradistinction to 10.1.5, the reference to Fido is here not by name (HBTR Sect. 3.4), but by pointer (HBTR Sect. 3.3).

For STAR-0 contents, the attributes R and 3rd are optional in that their values need only be defined if the content contains pro2 and pro3 pointers, respectively. For STAR-1 language contents (speak mode) and STAR-2 contents (hear mode), a value for the R attribute is obligatory; in small children, the R value may be viewed as fixed to the mother.

A rough idea of the location, of the date and the time of day, and of oneself will usually suffice for all practical purposes (Sect. 14.2). Loss of the STA parameter values for any length of time, however, will result in complete disorientation of the agent.

10.2 Speaker's STAR-1 Perspective on Stored Content

The next day (T: 1930-07-04), Simone and J.-P. meet in the Café de Flore on the Left Bank. To get the conversation going, J.-P. begins a *statement dialog* (10.6.7) by uttering the following declarative sentence to her (in French):

10.2.1 STAR-1 EXPRESSION WITH 1ST AND 2ND PERSON INDEXICALS

I wrote you a letter yesterday.$^{\text{STAR}-1}$

At this point, J.-P. is dealing with two STARs, (i) the STAR-0 defined in 10.1.3 and used in the anchored content 10.1.4 and (ii) the STAR-1 for the utterance indicated in 10.2.1. The S, A, and R values of these two STARs in J.-P.'s mind happen to be the same,[7] but the T values differ. This difference constitutes a second kind of perspective: J.-P. is looking back onto a content created in his mind in the recent past. The English surface of 10.2.1 reflects this perspective by means of (i) the past tense form of the finite verb and (ii) the adverbial modifier yesterday.

At the level of content, the automatic coding of the speaker perspective is based on the following DBS inference, called STAR-1.1. It takes an anchored STAR-0 content like 10.1.4 and the speaker's current STAR-1 as input:

10.2.2 STAR-1.1 INFERENCE FOR TEMPORAL SPECIFICATION

$$\begin{bmatrix}\text{verb: }\alpha\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: O}\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{S: L}'\\ \text{T: D}'\\ \text{A: N}\\ \text{R: O}'\\ \text{prn: K+M}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: }\alpha\\ \text{sem: }\beta\\ \text{mdr: }\gamma\\ \text{prn: K+M}\end{bmatrix}\begin{bmatrix}\text{adj: }\gamma\\ \text{mdd: }\alpha\\ \text{prn: K+M}\end{bmatrix}\begin{bmatrix}\text{S: L}'\\ \text{T: D}'\\ \text{A: N}\\ \text{R: O}'\\ \text{prn: K+M}\end{bmatrix}$$

If D < D′, then β = past, and if D diff D′ = 1 day, then γ = yesterday; and similarly for all the other possible temporal relations between a STAR-0 and a STAR-1 differing in their T value.

In the input pattern, an anchored STAR-0 content is represented by patterns for a verb and for a STAR-0 which share the prn variable K (first two proplets). The third input proplet is the agent's current STAR-0 with the prn value K+M. The output pattern re-anchors the STAR-0 of the content to the speaker's current STAR-0 with the prn value K+M, resulting in a STAR-1; its S, T, and R values L′(ocation), D′(ate), and O′(blique), respectively, may differ from those of the STAR-0, but the A value N(ominative) must be the same.

The output is represented by a modified pattern for the verb, an additional proplet pattern for an (optional) adverbial modifier, and the STAR-1 pattern. The restrictions on the variables β and γ are used to control the tense and the temporal adverbial specified in the output pattern of the inference.

Applying the inference STAR-1.1 to (i) J.-P.'s current STAR-1 and (ii) the STAR-0 content 10.1.4 results in the following STAR-1 content (speak mode):

10.2.3 STAR-1 CONTENT pro1 wrote pro2 a letter yesterday

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: write}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{verb: write}\\ \text{arg: pro1 pro2 letter}\\ \text{sem: past}\\ \text{mdr: yesterday}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: write}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{noun: letter}\\ \text{fnc: write}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{adj: yesterday}\\ \text{mdd: write}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-04}\\ \text{A: J.-P. Sartre}\\ \text{R: S. de Beauvoir}\\ \text{prn: 659+7}\end{bmatrix}$$

Compared to 10.1.4, the sem attribute of the verb has received the value past, and the adverbial modifier yesterday has been added as a proplet connected to the verb. Also, the new prn value 659+7 has been assigned by the parser not just to the verb and adj proplets matched by the output pattern of the STAR-1.1 inference, but to all proplets of the resulting content.

The output of the inference is written to the now front of the agent's word bank. Thus, the original content and its STAR-0 are not overwritten. They may be retrieved by using the first part of the new content's prn value, here 659.

This is in concord with the content-addressable database of a word bank, in which content is written once and never changed (Sects. 4.1, 14.5, 14.6). The

[7] The difference in location between J.-P.'s apartment and the Café de Flore within Paris is not reflected by the STARs' S values. Also, whole days are used for the T values. This *soft* treatment of the spatio-temporal coordinates (Sect. 14.2) is more appropriate for modeling a natural agent's cognition than are the unnaturally precise values provided by the natural sciences, though they may be used if needed.

antecedent of a DBS inference may only *read* content, while the consequent may only *write* to the now front of the word bank – as reflected by the prn variables of the STAR inferences 10.2.2 and 10.2.4.

In addition to the temporal respecification provided by the STAR-1.1 inference there must be a STAR-1.2 inference for a possible respecification of the S value and a STAR-1.3 inference for a possible respecification of the R value. For example, if J.-P. were to meet Juliette instead of Simone at the Café de Flore, the R value of his STAR-1 would be Juliette and the content of 10.1.4 would be realized as I wrote Simone a letter yesterday. The STAR-1.3 inference is defined as follows:

10.2.4 STAR-1.3 INFERENCE FOR SPECIFICATION OF RECIPIENT

$$\begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\{\text{X pro2}\}\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: O}\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{S: L}'\\ \text{T: D}'\\ \text{A: N}\\ \text{R: O}'\\ \text{prn: K+M}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\{\text{X O}\}\\ \text{prn: K+M}\end{bmatrix}\begin{bmatrix}\text{noun: O}\\ \text{fnc: }\alpha\\ \text{prn: K+M}\end{bmatrix}\begin{bmatrix}\text{S: L}'\\ \text{T: D}'\\ \text{A: N}\\ \text{R: O}'\\ \text{prn: K+M}\end{bmatrix}$$

The notation {X pro2} is intended to indicate that pro2 and X (for other category segments) will match content counterparts in any order. In this way, the inference may result in contents like I wrote Simone a letter as well as Simone wrote me a letter.

The first three pattern proplets of the antecedent match the STAR-0 content 10.1.4. The consequent (output pattern) replaces pro2 by the R value O of the STAR-0 and assigns the prn value of the STAR-1.

Applying STAR-1.1 and STAR-1.3 to 10.1.4 results in the following content:

10.2.5 STAR-1 CONTENT pro1 wrote Simone a letter yesterday

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: write}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{verb: write}\\ \text{arg: pro1 Simone letter}\\ \text{sem: past}\\ \text{mdr: yesterday}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{noun: Simone}\\ \text{fnc: write}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{noun: letter}\\ \text{fnc: write}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{adj: yesterday}\\ \text{mdd: write}\\ \text{prn: 659+7}\end{bmatrix}\begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-04}\\ \text{A: J.-P. Sartre}\\ \text{R: Juliette}\\ \text{prn: 659+7}\end{bmatrix}$$

The derivation of a STAR-1 content does not interfer with the communication cycle because the lexicalization rules of the speak mode are integrated into the proplet patterns matching the content (4.6.2, 4.6.3; NLC Sects. 12.4–12.6).

10.3 Hearer's STAR-2 Perspective on Language Content

When Simone hears the utterance 10.2.1, she does a standard time-linear DBS.Hear derivation, resulting in the following set of proplets:

10.3.1 Result of analyzing 10.2.1 in the hear mode

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: write}\\ \text{prn: 623}\end{bmatrix} \begin{bmatrix}\text{verb: write}\\ \text{arg: pro1 pro2 letter}\\ \text{sem: past}\\ \text{mdr: yesterday}\\ \text{prn: 623}\end{bmatrix} \begin{bmatrix}\text{noun: pro2}\\ \text{fnc: write}\\ \text{prn: 623}\end{bmatrix} \begin{bmatrix}\text{noun: letter}\\ \text{fnc: write}\\ \text{prn: 623}\end{bmatrix} \begin{bmatrix}\text{adj: yesterday}\\ \text{mdd: write}\\ \text{prn: 623}\end{bmatrix}$$

As the result of a surface compositional reconstruction of the speaker's literal meaning$_1$, the content 10.3.1 represents the perspective of the speaker J.-P. – except for the prn value 623, which equals that of Simone's current STAR-0.

In the hear mode, the following main perspectives[8] on incoming speak mode content must be distinguished:

10.3.2 Main hear mode perspectives on language content

1. The perspective of the hearer as the partner in face-to-face communication.
2. The perspective of someone overhearing a conversation between others.
3. The reader's perspective onto the content of a written text (Chap. 11).

Given that Simone and J.-P. are partners in face-to-face communication, the correct way for Simone to convert J.-P.'s speak mode perspective is from

I wrote you a letter yesterday.$^{\mathrm{STAR-1}}$

to her own hear mode perspective as

You wrote me a letter yesterday.$^{\mathrm{STAR-2}}$

This automatic conversion is based on the STAR-2.1 inference (see 10.3.3 below) which takes a content like 10.3.1 and a STAR-1 as input. The STAR-1 is *attributed* by the hearer to the speaker. In face-to-face communication, this is easy because the speaker's STAR-1 and the hearer's STAR-2 correspond in that their S and T values are the same, and their A and R values are reversed. The DBS inference STAR-2.1 is defined as follows:

10.3.3 STAR-2.1 inference for deriving hearer perspective

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\{\text{X pro1 pro2}\}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: pro2}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: O}\\ \text{prn: K}\end{bmatrix} \Rightarrow$$

$$\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\{\text{X pro2 pro1}\}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: pro1}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: O}\\ \text{R: N}\\ \text{prn: K}\end{bmatrix}$$

8 Special cases are phone conversations which require the hearer to recompute the speaker's S(pace) value; when talking across time zones, the speaker's T(ime) value must be recomputed as well.

In the output, the speaker's STAR-1 perspective is revised into the hearer's current STAR-2 perspective by swapping the A and R values N and O, and keeping the S, T, and prn values. The notation {X pro1 pro2} is intended to indicate that pro1, pro2, and X (for other category segments) will match content counterparts in any order.

If we assume that the STAR-2 used by Simone is [S: Paris], [T: 1930-07-04], [A: Simone], and [R: J.-P.], then the application of the inference STAR-2.1 to content 10.3.1 results in the following STAR-2 content:

10.3.4 STAR-2 CONTENT pro2 wrote pro1 a letter yesterday

$$\begin{bmatrix}\text{noun: pro2}\\\text{fnc: write}\\\text{prn: 623}\end{bmatrix}\begin{bmatrix}\text{verb: write}\\\text{arg: pro2 pro1 letter}\\\text{sem: past}\\\text{mdr: yesterday}\\\text{prn: 623}\end{bmatrix}\begin{bmatrix}\text{noun: pro1}\\\text{fnc: write}\\\text{prn: 623}\end{bmatrix}\begin{bmatrix}\text{noun: letter}\\\text{fnc: write}\\\text{prn: 623}\end{bmatrix}\begin{bmatrix}\text{adj: yesterday}\\\text{mdd: write}\\\text{prn: 623}\end{bmatrix}\begin{bmatrix}\text{S: Paris}\\\text{T: 1930-07-04}\\\text{A: S. de Beauvoir}\\\text{R: J.-P. Sartre}\\\text{prn: 623}\end{bmatrix}$$

Here, pro2 is pointing at the R value J.-P., pro1 is pointing at the A value Simone, yesterday is pointing at the T value 1930-07-04, and the sem attribute of the verb has the value past from the hear mode analysis of the surface.

As an example without any indexicals consider Simone producing the following utterance addressed to J.-P., using her STAR-0 content 10.1.5:

10.3.5 STAR-1 CONTENT WITHOUT INDEXICALS

Fido barked.$^{\text{STAR}-1}$

For the hearer J.-P., the interpretation of this content requires adjusting Simone's STAR-1 speak mode perspective to J.-P.'s STAR-2 hear mode perspective. This is based on the following STAR-2.2 inference:

10.3.6 STAR-2.2 INFERENCE FOR CONTENT WITHOUT INDEXICALS

$$\begin{bmatrix}\text{verb: }\alpha\\\text{prn: K}\end{bmatrix}\begin{bmatrix}\text{S: L}\\\text{T: D}\\\text{A: N}\\\text{R: O}\\\text{prn: K}\end{bmatrix}\Rightarrow\begin{bmatrix}\text{verb: }\alpha\\\text{prn: K}\end{bmatrix}\begin{bmatrix}\text{S: L}\\\text{T: D}\\\text{A: O}\\\text{R: N}\\\text{prn: K}\end{bmatrix}$$

J.-P. assigns Simone's STAR-1 to the input by swapping the A and R values of his STAR-2, coding J.-P.'s perspective, with the following result:

10.3.7 STAR-2 CONTENT Fido barked

$$\begin{bmatrix}\text{noun: Fido}\\\text{fnc: bark}\\\text{prn: 572}\end{bmatrix}\begin{bmatrix}\text{verb: bark}\\\text{arg: Fido}\\\text{sem: past}\\\text{prn: 572}\end{bmatrix}\begin{bmatrix}\text{S: Paris}\\\text{T: 1930-07-03}\\\text{A: J.-P. Sartre}\\\text{R: S. de Beauvoir}\\\text{prn: 572}\end{bmatrix}$$

The properties of STAR-2 contents resulting from STAR-1 contents transmitted in face-to-face communication may be summarized as follows:

10.3.8 Properties of STAR-2 inferences

1. The S value of the STAR-1 in the input (matching the antecedent) equals the S value of the STAR-2 in the output (derived by the consequent).
2. The T value of the STAR-1 in the input equals the T value of the STAR-2 in the output.
3. The A value of the STAR-1 in the input equals the R value of the STAR-2 in the output.
4. The R value of the STAR-1 in the input equals the A value of the STAR-2 in the output.
5. The prn value of the input equals the prn value of the output.

These properties hold specifically for STAR-2 contents. For example, in STAR-1 contents the A and the R values are not inverted by an inference. The derivation of a STAR-2 content does not interfer with the communication cycle because the lexicalization rules of the speak mode are integrated into the proplet patterns matching the content (4.6.2, 4.6.3; NLC Sects. 12.4–12.6).

10.4 Dialog with a WH Question and Its Answer

The statement dialog analyzed in Sects. 10.1–10.3 consists of the speaker producing and the hearer interpreting a declarative[9] sentence. A question dialog (10.6.7), in contrast, is based on (1) the questioner producing an interrogative, (2) the answerer interpreting the interrogative, (3) the answerer producing an answer, and (4) the questioner interpreting the answer (10.6.7, 2).

Preceding these four steps, however, there is the emergence of the question *content*. For example, having digested J.-P.'s remark 10.2.1, Simone searches her recent memory for connected letter, write, and J.-P. proplets, and realizes

[9] The sentential mood associated with statement dialogs is the *declarative*, just as the sentential mood associated with question dialogs is the *interrogative*, while the sentential mood is associated with requests is the *imperative*. However, the sentential moods are often used indirectly. For example, the Yes/No interrogative Could you pass the salt? is normally used as as a request rather than a question. Similarly, I demand that signed statement, now. (declarative as indirect request) is answered in the movie Bullit (Yates 1968) with the politeness formula Excuse me. uttered by the addressee walking away (indirect way of refusing the request). For simplicity, statement, question, and request dialogs are illustrated here with literally used expressions in the declarative, interrogative, and imperative moods, respectively.

that she has not yet received the letter.[10] This creates a certain kind of imbalance in her mind, commonly known as curiosity. As a means to regain her equilibrium, the following question content emerges in Simone's mind:

10.4.1 Nonlanguage content in the interrogative mood

What did you write?$^{\text{STAR}-0}$

In analogy to 10.1.1–10.1.4, the content and its anchor may be represented as the following set of proplets:

10.4.2 Anchored STAR-0 content of WH interrogative

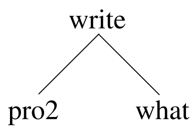
$$\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: write}\\ \text{prn: 625}\end{bmatrix} \begin{bmatrix}\text{verb: write}\\ \text{cat: \#n}' \text{ \#a}' \text{ interrog}\\ \text{sem: past}\\ \text{arg: pro2 what}\\ \text{prn: 625}\end{bmatrix} \begin{bmatrix}\text{noun: what}\\ \text{fnc: write}\\ \text{prn: 625}\end{bmatrix} \begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-04}\\ \text{A: S. de Beauvoir}\\ \text{R: J.-P. Sartre}\\ \text{prn: 625}\end{bmatrix}$$

As in 10.1.4, the connection between the STAR and the content is established by a common prn value, here 625. The indexical pro2 points at the R value of the STAR-0.

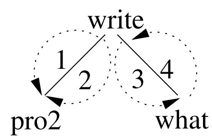
Given that there is no significant time difference between the formation of the content and its use for language production, there is no need to derive a separate speaker perspective on the content (in contradistinction to the transition from 10.1.1 to 10.2.1). Instead, Simone proceeds to realize the surface in 10.4.1 by using (i) 10.4.2 as a STAR-1 content and (ii) the following DBS graph analysis:

10.4.3 Questioner as speaker: DBS graph analysis of 10.4.1

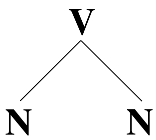
(i) SRG (semantic relations graph)



(ii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

3	4	1	2
What	did	you	write_?
V\N	N\V	V/N	N/V

[10] Despite the much praised postal service by pneumatic delivery in Paris at the time (Beauvoir 1960).

As shown in Sect. 7.4, a NAG and a proplet representation of a content (here 10.4.2) are sufficient for quasi-automatically deriving a DBS.Speak grammar which realizes a corresponding surface.

Now it is J.-P.'s turn to interpret the incoming surface (presented here in English) in the hear mode:

10.4.4 ANSWERER AS HEARER PARSING 10.4.1

lexical lookup

What	did	you	write	?
noun: wh? cat: np3 sem: −hum fnc: prn:	verb: d_1 cat: n' do' v sem: past arg: prn:	noun: pro2 cat: sp2 sem: fnc: prn:	verb: write cat: n−s3' a' v sem: pres arg: prn:	verb: ? cat: vi' interrog prn:

syntactic−semantic parsing

1	noun: wh? cat: np3 sem: −hum fnc: prn: 668	verb: d_1 cat: n' do' vi sem: past arg: prn:	*cross−copying*

2	noun: wh? cat: np3 sem: −hum fnc: d_1 prn: 668	verb: d_1 cat: n' #a' vi sem: past arg: wh? prn: 668	noun: pro2 cat: sp2 sem: fnc: prn:	*cross−copying*

3	noun: wh? cat: np3 sem: −hum fnc: d_1 prn: 668	verb: d_1 cat: #n' #a' vi sem: past arg: pro2 wh? prn: 668	noun: pro2 cat: sp2 sem: fnc: d_1 prn: 668	verb: write cat: n−s3' a' v sem: pres arg: prn:	*absorption with simultaneous substitution*

4	noun: wh? cat: np3 sem: −hum fnc: write prn: 668	verb: write cat: #n' #a' vi sem: past arg: pro2 wh? prn: 668	noun: pro2 cat: sp2 sem: fnc: write prn: 668	verb: ? cat: vi' interrog prn:	*absorption*

result

noun: wh? cat: np3 sem: −hum fnc: write prn: 668	verb: write cat: #n' #a' interrog sem: past arg: pro2 wh? prn: 668	noun: pro2 cat: sp2 sem: fnc: write prn: 668

The result of this strictly time-linear, surface compositional derivation is a content which represents the perspective of the speaker Simone – except for the prn value, here 668, which is assigned by the hearer J.-P.

For understanding, the answerer J.-P. must change Simone's speak mode perspective into his own hear mode perspective by transforming the content pro2 wrote what?[11] of Simone's question into pro1 wrote what?, based on the following STAR-2.3 inference:

10.4.5 STAR-2.3 INFERENCE FOR DERIVING HEARER PERSPECTIVE

$$
\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: Z interrog}\\ \text{arg: \{X pro2 wh?\}}\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: wh?}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: O}\\ \text{prn: K}\end{bmatrix}
\Rightarrow
$$

$$
\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: Z interrog}\\ \text{arg: \{ X pro1 wh?\}}\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: wh?}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: O}\\ \text{R: N}\\ \text{prn: K}\end{bmatrix}
$$

This STAR-2 inference complies with 10.3.8. Applying it to (i) the result of the hear mode derivation 10.4.4 and (ii) J.-P.'s current STAR-2 produces the following anchored content:

10.4.6 RESULT OF APPLYING THE STAR-2.3 INFERENCE TO 10.4.4

$$
\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: write}\\ \text{prn: 668}\end{bmatrix}
\begin{bmatrix}\text{verb: write}\\ \text{cat: \#n}'\text{ \#be}'\text{ interrog}\\ \text{sem: past}\\ \text{arg: pro1 wh?}\\ \text{prn: 668}\end{bmatrix}
\begin{bmatrix}\text{noun: wh?}\\ \text{fnc: write}\\ \text{prn: 668}\end{bmatrix}
\begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-04}\\ \text{A: J.-P. Sartre}\\ \text{R: S. de Beauvoir}\\ \text{prn: 668}\end{bmatrix}
$$

At this point, the answerer J.-P. understands Simone's question. This has the effect of passing Simone's original imbalance successfully on to J.-P. as the hearer. To reestablish his equilibrium, J.-P. searches his recent memory for connected letter, write, and Simone proplets. When he finds the answer, J.-P. uses the speak mode to reply as follows (in French):

10.4.7 ANSWERER AS SPEAKER

A little poem.$^{\text{STAR}-1}$

The content underlying this answer has the following proplet representation:

10.4.8 ANSWER TO A WH QUESTION AS A SET OF STAR-0 PROPLETS

$$
\begin{bmatrix}\text{noun: poem}\\ \text{sem: indef sg}\\ \text{mdr: little}\\ \text{prn: 655}\end{bmatrix}
\begin{bmatrix}\text{adj: little}\\ \text{sem: pad}\\ \text{mdd: poem}\\ \text{prn: 655}\end{bmatrix}
\begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-03}\\ \text{A: J.-P. Sartre}\\ \text{R: S. de Beauvoir}\\ \text{prn: 655}\end{bmatrix}
$$

---

[11] The result of a hear mode derivation is a *set* of proplets, i.e. order-free. In other words, the sets *pro2 wrote what?* and *What wrote pro2?* are equivalent.

At the level of content, a pertinent answer must *precede* the question in the mind of the answerer.[12] It is only when the answer is realized in language that it follows the question, for answerer and questioner alike.

The final turn of a question-answer sequence is the questioner in the hear mode. In our example, Simone as the hearer parses J.-P.'s answer as follows:

10.4.9 QUESTIONER AS HEARER PARSING 10.4.7

	A	little	poem	•	
lexical lookup	noun: n_1 cat: sn' snp fnc: mdr: prn:	adj: little cat: adn mdd: mdr: prn:	noun: poem cat: sn fnc: mdr: prn:	verb: • cat: v' decl prn:	
syntactic−semantic parsing					
1	noun: n_1 cat: sn' snp fnc: mdr: prn: 626	adj: little cat: adn mdd: mdr: prn:			*cross−copying*
2	noun: n_1 cat: sn' snp fnc: mdr: little prn: 626	adj: little cat: adn mdd: n_1 mdr: prn: 626	noun: poem cat: sn fnc: mdr: prn:		*absorption with simultaneous substitution*
3	noun: poem cat: snp fnc: mdr: little prn: 626	adj: little cat: adn mdd: poem mdr: prn: 626		verb: • cat: v' decl prn:	*absorption*
result	noun: poem cat: snp fnc: decl mdr: little prn: 626	adj: little cat: adn mdd: poem mdr: prn: 626			

On the one hand, the answer to a WH question is in the declarative mood, as indicated by the period. On the other hand, it is missing a verb. To characterize the result of the above derivation as a WH answer, we propose copying the value decl of the period into the fnc slot of the noun (line 3), and using the [fnc: decl] feature in the result as a marker for WH answers. This characterizes WH answers uniquely and allows easy definition of a proplet pattern for their retrieval.

While balance is reestablished for the answerer when uttering the answer (10.4.7), the questioner must not only interpret the answer, as in 10.4.9, but

[12] This is reflected by the prn values 668 of the question 10.4.6 and 655 of the answer 10.4.8.

combine the WH question and the answer into one declarative content. Derived by the following STAR-2 inference, the resulting content has a new prn value:

10.4.10 STAR-2.4 CONNECTING WH ANSWER TO INTERROGATIVE

$$\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: Z interrog}\\ \text{arg: }\{\text{X pro2 wh?}\}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: wh?}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: O}\\ \text{prn: K}\end{bmatrix} + \begin{bmatrix}\text{noun: }\beta\\ \text{fnc:decl}\\ \text{mdr: }\gamma\\ \text{prn: K+M}\end{bmatrix} \begin{bmatrix}\text{adj: }\gamma\\ \text{mdd: }\beta\\ \text{prn: K+M}\end{bmatrix} \Rightarrow$$

$$\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: Z decl}\\ \text{arg: }\{\text{ X pro2 }\beta\}\\ \text{prn: K+M}\end{bmatrix} \begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\alpha\\ \text{mdr: }\gamma\\ \text{prn: K+M}\end{bmatrix} \begin{bmatrix}\text{adj: }\gamma\\ \text{mdd: }\beta\\ \text{prn: K+M}\end{bmatrix} \begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: O}\\ \text{prn: K+M}\end{bmatrix}$$

The inference fills the WH gap in the interrogative by replacing the *what* pattern proplet with the noun of the answer, including an optional pattern proplet for an adnominal modifier. The cat value interrog in the input verb is changed to decl in the output. The func value decl of the input answer noun is changed to the core value of the verb in the output. The result of applying this inference to the inputs 10.4.2 and 10.4.8 is as follows:

10.4.11 QUESTIONER'S STAR-2 CONTENT FOR REGAINING BALANCE

$$\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: write}\\ \text{prn: 625+2}\end{bmatrix} \begin{bmatrix}\text{verb: write}\\ \text{cat: \#n}'\text{ \#do}'\text{ decl}\\ \text{sem: past}\\ \text{arg: pro2 poem}\\ \text{prn: 625+2}\end{bmatrix} \begin{bmatrix}\text{noun: poem}\\ \text{fnc: write}\\ \text{mdr: little}\\ \text{prn: 625+2}\end{bmatrix} \begin{bmatrix}\text{adj: little}\\ \text{mdd: poem}\\ \text{prn: 625+2}\end{bmatrix} \begin{bmatrix}\text{S: Paris}\\ \text{T: 1930-07-04}\\ \text{A: S. de Beauvoir}\\ \text{R: J.-P. Sartre}\\ \text{prn: 625+2}\end{bmatrix}$$

The input prn value 625 is incremented by 2 because between Simone's production of the question content 10.4.2 and the inferencing deriving the content of 10.4.11 there is Simone's interpretation of J.-P.'s answer, as shown in 10.4.9.

10.5 Dialog with a Yes/No Question and Its Answer

WH questions may request noun values, as in What did you write?, or adjective values, as in How did you sleep? (manner), Why did you go? (cause), When did you leave? (time), and Where are you now? (location). They all conform to the time-linear structure of question dialogs (10.6.7), and their DBS analysis closely resembles that shown in the previous section for a WH question with a noun answer.

Unlike WH questions, Yes/No questions request a choice between only two values, namely yes and no, in their various guises. For example, after J.-P.'s answer A little poem., Simone's curiosity is not completely satisfied. As a potential countermeasure, the following STAR-0 content emerges in her mind:

10.5.1 STAR-0 content underlying language countermeasure

Is the poem about me? $^{\text{STAR}-0}$

As in 10.4.1 and 10.4.2, this content is used by Simone as a STAR-1 content and realized as a surface (in French). From this unanalyzed external surface the answerer J.-P. derives the following content:

10.5.2 Answerer as hearer parsing a Yes/No interrogative

Is	the	poem	about	me	?
lexical lookup					
verb: v_1 cat: sn3' be' v sem: pres arg: prn:	noun: n_1 cat: nn' np sem: def fnc: prn:	noun: poem cat: sn sem: sg fnc: prn:	noun: n_2 cat: adnv np' sem: *about* mdd: prn:	noun: pro1 cat: obl sem: sg fnc: prn:	verb: ? cat: vi' interrog prn:

syntactic−semantic parsing

1	verb: v_1 cat: sn3' be' vi sem: pres arg: prn: 669	noun: n_1 cat: nn' np sem: def fnc: prn:				*cross−copying*
2	verb: v_1 cat: #sn3' be' vi sem: pres arg: n_1 prn: 669	noun: n_1 cat: nn' np sem: def fnc: v_1 prn: 669	noun: poem cat: sn sem: sg fnc: prn:			*absorption with simultaneous substitution*
3	verb: v_1 cat: #sn3' be' vi sem: pres arg: poem prn: 669	noun: poem cat: snp sem: def sg fnc: v_1 prn: 669		noun: n_2 cat: adnv np' sem: *about* mdd: prn:		*cross−copying*
4	verb: v_1 cat: #sn3' #be' vi sem: pres arg: poem n_2 prn: 669	noun: poem cat: snp sem: def sg fnc: v_1 prn: 669		noun: n_2 cat: adnv np' sem: *about* fnc: v_1 prn: 669	noun: pro1 cat: obl sem: sg fnc: prn:	*absorption with simultaneous substitution*
5	verb: v_1 cat: #sn3 #be vi sem: pres arg: poem pro1 prn: 669	noun: poem cat: snp sem: def sg fnc: v_1 prn: 669		noun: pro1 cat: adnv obl sem: *about* sg fnc: v_1 prn: 669	verb: ? cat: vi' interrog prn:	*absorption*

result

verb: v_1	noun: poem	noun: pro1
cat: #sn3’ #be’ interrog	cat: snp	cat: adnv obl
sem: pres	sem: def sg	sem: *about* sg
arg: poem pro1	fnc: v_1	fnc: v_1
prn: 669	prn: 669	prn: 669

The example shows the use of the auxiliary be as a main verb (copula, NLC 6.6.9–6.6.11). The phrase about me is used as a prepositional object which cancels the be′ valency position in the category of the copula (lines 3 and 4). The preposition *about*[13] is coded as the first value in the sem slot of the prepositional object (line 4). The result codes the perspective of the questioner Simone as the speaker, except for the prn value, here 669, which is assigned by the answerer J.-P. as the hearer.

Before storage in J.-P.'s word bank, Simone's speaker perspective coded as poem is about pro1? must be revised into the perspective of the answerer as hearer, coded as poem is about pro2? and represented as follows:

10.5.3 Answerer as hearer: revised perspective of 10.5.2

verb: v_1	noun: poem	noun: pro2	S: Paris
cat: #sn3′ #be′ interrog	cat: snp	cat: adnv obl	T: 1930-07-04
sem: pres	sem: indef sg	sem: *about* sg	A: J.-P. Sartre
arg: poem pro2	fnc: v_1	fnc: v_1	R: S. de Beauvoir
prn: 669	prn: 669	prn: 669	prn: 669

The revision of perspective, i.e. the change of pro1 into pro2 and the concomitant exchanges of the A and R values in the STAR-2, is based on a STAR-2 inference like 10.4.5.

At this point, J.-P. understands Simone's question and experiences an imbalance similar to the one which caused Simone to ask the question in the first place. To find the answer and reestablish his homeostasis, J.-P. replaces the prn value 669 with a variable, e.g. K. This turns the revised STAR-2 content 10.5.3 into a proplet pattern which allows J.-P. to search his recent memory for a matching content (Sect. 6.5, automatic pattern derivation.)

A successful retrieval triggers a positive answer in J.-P.'s mind and switches the answerer from a hearer to a speaker. After stoking his pipe, J.-P. replies:

10.5.4 Answerer J.-P. as speaker

Yes.$^{\mathrm{STAR-1}}$

[13] Prepositions are function words like determiners (7.2.5). However, compared to the determiner values exh, sel, def, indef, sg, and pl (NLC 6.4.7), the number of prepositions is open-ended and their meaning is more complex and varied . To distinguish between prepositions like *about, above, before, below, in, on*, etc., their English surface is written into the first slot of the sem attribute in *italic* font.

With this utterance, the answerer regains his equilibrium, but Simone must still do a hear mode derivation (questioner as hearer) and combine the result with the anchored content 10.5.1 as a STAR-1 content, represented as a set of proplets. This is based on a STAR-2 inference like 10.4.10; it replaces the cat value interrog of the verb with decl and assigns a new prn value.

10.6 Dialog with a Request and Its Fulfillment

The third kind of basic dialog besides statement and question is the request dialog. For example, Simone notes that her current cigarette is about to be finished. Her need to dispose of the stub brings forth a slight imbalance, causing the following STAR-0 content to emerge as a countermeasure:

10.6.1 Anchored nonlanguage request content

(Please)[14] pass the ashtray! $^{\text{STAR}-0}$

Such a countermeasure may be learned as a one-step inference chain like 5.2.3.
The STAR-0 content 10.6.1 is represented as the following set of proplets:

10.6.2 Request STAR-0 content as a set of proplets

verb: pass	noun: ashtray	S: Paris
cat: vimp	cat: snp	T: 1930-07-04
sem: pres	sem: def sg	A: S. de Beauvoir
arg: ∅ ashtray	fnc: pass	R: J.-P. Sartre
prn: 630	prn: 630	prn: 630

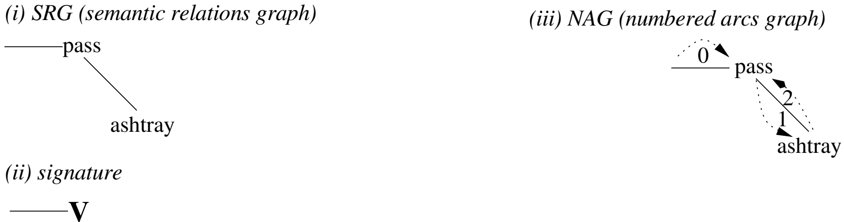
The requestor equals the A value Simone and the requestee equals the R value J.-P. The verb's cat value vimp, for imperative, shows the sentential mood.[15]

After a long last drag on her cigarette, Simone produces the surface in 10.6.1 based on the proplet set 10.6.2 and the following DBS graph structure:

[14] For simplicity, we omit the analysis of please. Its pragmatic role is to indicate a polite attitude towards the hearer, similarly to the use of vous instead of toi. Syntactico-semantically, please may be treated either as an adverbial modifier or be integrated into a general treatment of *particles* (Ickler 1994). The widespread natural language phenomenon of particles may require a treatment in terms of pragmatics which is relatively independent of the semantic relations of structure, i.e. functor-argument and coordination.

[15] The value **vimp** (for imperative form of the verb) must be distinguished **imp** (for imperfect) and **impl** (for implies).

10.6.3 Graph structure used by requestor as speaker



(iv) surface realization

0	1	2
Pass	the_ashtray	!
V/V	V/N	N/V

Production of the content 10.6.1 as a surface constitutes step 1 of the request-fulfillment sequence. Given that there is no significant time difference between the formation of the content and its use for language production, there is no need to derive a separate speaker perspective. In other words, Simone may reuse the STAR-0 of the content 10.6.2 as the STAR-1 of her utterance.

In step 2 (requestee as hearer), J.-P. parses the surface in 10.6.1 as follows:

10.6.4 Requestee as hearer parsing Pass the ashtray!

lexical lookup

Pass	the	ashtray	!
verb: pass cat: n−s3' a' v sem: pres arg: prn:	noun: n_1 cat: nn' np sem: def fnc: prn:	noun: ashtray cat: sn sem: sg fnc: prn:	verb: ! cat: vimp' impv prn:

syntactic−semantic parsing

1	verb: pass cat: n−s3' a' v sem: pres arg: Ø prn: 671	noun: n_1 cat: nn' np sem: def fnc: prn:		*cross−copying*
2	verb: pass cat: #a' vimp sem: pres arg: Ø n_1 prn: 671	noun: n_1 cat: nn' np sem: def fnc: pass prn: 671	~~noun:~~ ashtray ~~cat: sn~~ ~~sem:~~ sg fnc: prn:	*absorption with simultaneous substitution*
3	verb: pass cat: #a' vimp sem: pres arg: Ø ashtray prn: 671	noun: ashtray ~~cat: snp~~ sem: def sg fnc: pass prn: 671	verb: ! ~~cat: v~~imp' impv prn:	*absorption*

result

verb: pass	noun: ashtray
cat: #a' impv	cat: snp
sem: pres	sem: def sg
arg: ∅ ashtray	fnc: pass
prn: 671	prn: 671

The lexical analysis of pass is that of the standard unmarked present tense form of a verb (citation form in English).[16]

The conversion of the lexical cat value n-s3′ a′ v to a′ vimp is accomplished syncategorematically by the DBS.Hear operation which combines a sentence-initial finite verb with a noun (NLC 14.6.4). The result of this hear mode derivation equals the content 10.6.2 of the requestor as speaker, except that the hearer routinely assigns a new prn value and a new STAR-2:

10.6.5 Request STAR-2 content as a set of proplets

verb: pass	noun: ashtray	S: Paris
cat: #a vimp	cat: snp	T: 1930-07-04
sem: pres	sem: def sg	A: J.-P. Sartre
arg: ∅ ashtray	fnc: pass	R: S. de Beauvoir
prn: 671	prn: 671	prn: 671

Here the conversion from Simone's perspective as the requestor to J.-P.'s perspective as the requestee is limited to exchanging the A and R values in the STAR-1 (= STAR-0) of 10.6.2. This has an effect similar to going from pro2 pass ashtray to pro1 pass ashtray, though the imperative content has no explicit subject, and consequently no indexical.

At this point, J.-P. understands the request: the slight imbalance experienced by Simone has been successfully passed on to him by means of natural language. As step 3 of a time-linear request sequence, the requestee must take the requested action in order to be cooperative. This may be a nonlanguage action, as responding to Open the window!, or a language action, as responding to Tell me more about your mother! (Weizenbaum 1965).

[16] The main sentential moods in natural language are the declarative, the interrogative, and the imperative. These sentential moods must be distinguished from the verbal moods, i.e. the indicative, the subjunctive, and the imperative in grammars for English (Sect. 5.6). Thus traditional grammar uses the term "imperative" ambiguously for a sentential and a verbal mood.

In some languages the verbal moods are realized as special forms of the verbal morphology, with dedicated uses in the associated sentential moods. Classical Latin, for example, has separate word forms for the verb in the imperative mood, differentiated by whether the requestee is a singular individual (curre, *run* vimp sg) or a plural group (currete, *run* vimp pl).

English, in contrast, uses the unmarked form of the verb's present tense for constructing the imperative as a sentential mood. Thus, just as there is no separate verbal form for the infinitive (6.6.5–6.6.7, Sect. 15.4), there is no separate verbal form for the imperative. Instead, the imperative as a sentential mood is built as a special syntactic-semantic construction, characterized by word order, intonation, or interpunctuation, and the absence of a subject, a tense, and a verbal mood.

In our example, the requestee J.-P. has been asked to perform a nonlanguage action. Being cooperative, he is looking around for a means to realize his blueprint for action, which may be paraphrased as pro1 pass ashtray. He recognizes the ashtray on the restaurant table (the year is 1930) and initiates a manipulation sequence which moves the ashtray over to Simone. This results in J.-P. regaining his equilibrium.

The final step 4 is the requestor recognizing the fulfillment of the request by the requestee. In our case, Simone puts out her cigarette in the ashtray. This results in her regaining balance and closes the sequence.

The main difference between the time-linear sequence of a question and a request dialog is that steps 3 and 4 in a question dialog are normally realized as language action, while the corresponding steps in a request dialog may be either language or nonlanguage action. However, because all essential dialog operations, i.e. adjustments of perspective and other inferencing, are performed at the level of content, DBS can handle request dialogs just as well as question dialogs.

In summary, this chapter has analyzed dialog as a sequence of elementary dialogs, using a fictional conversation between Jean-Paul Sartre and Simone de Beauvoir in the Café de Flore on 1930-07-04 as our example:

10.6.6 Characteristic sequences of elementary dialogs

J.-P. Sartre:	I wrote you a letter yesterday.	(statement, Sect. 10.1–10.3)
S. de Beauvoir:	What did you write?	(WH question, Sect. 10.4)
J.-P. Sartre:	A little poem.	(WH answer, Sect. 10.4)
S. de Beauvoir:	Is the poem about me?	(Yes/No question, Sect. 10.5)
J.-P. Sartre :	Yes.	(Yes/No answer, Sect. 10.5)
S. de Beauvoir:	(Please) pass the ashtray!	(request, Sect. 10.6)
J.-P. Sartre:	passes ashtray	(fulfillment, Sect. 10.6)

Each of the elementary dialogs is a characteristic sequence of turns which consist of the following perspective conversions:

10.6.7 Perspective conversions as time-linear sequences

1. Statement Dialog
   STAR-0: emergence of a nonlanguage content in agent A (Sect. 10.1)
   STAR-1: production of a statement by agent A as the speaker (Sect. 10.2)
   STAR-2: interpretation of the statement by agent B as the hearer (Sect. 10.3)

2. Question Dialog (WH Sect. 10.4, Yes/No Sect.10.5)
   STAR-0: emergence of a nonlang. content in agent A as the questioner
   STAR-1: production of a question by agent A as the speaker
   STAR-2: interpretation of the question by agent B as the hearer
   STAR-1: production of an answer by agent B as the speaker
   STAR-2: interpretation of the answer by agent A as the hearer

3. Request Dialog (Sect. 10.6)
   STAR-0: emergence of a nonlanguage content in agent A as the requestor
   STAR-1: production of a request by agent A as the speaker
   STAR-2: interpretation of the request by agent B as the hearer
   STAR-1: nonlang. or lang. fulfillment action by agent B as the requestee
   STAR-2: nonlanguage or language fulfillment recognition by agent A as the requestor

Our analysis has proceeded systematically from the perspective of the speaker to that of the hearer. The transition from one perspective to the next is described in terms of explicit STAR inferences. Part of the analysis is the interpretation of first and second person pronouns (indexicals), which is different for the speaker and the hearer. Our theory complements Schegloff's (2007) sociolinguistic analysis of many recorded and transcribed dialogs.

11. Computing Perspective in Text

Natural language surfaces transport three basic kinds of information. They are (i) the propositional content, or content for short, (ii) the evaluation of the content, also called appraisal, and (iii) the perspective of the speaker onto content regarding spatio-temporal anchoring (14.2) and the interpretation of first, second, and third person pronouns. The transfer of (i), propositional content, has been discussed throughout the previous chapters. The mechanism of appraisal (ii) for the purpose of maintaining balance has been focused on in Chap. 5. In this chapter we continue with (iii): having analyzed the computation of perspective in spoken dialog (Chap. 10) we turn now to computing perspective in written text.

11.1 Coding the STAR-1 in Written Text

It holds in general that language signs are produced and interpreted relative to their *origin* (FoCL 5.3.4). This means that (i) the speaker's perspective from the production situation (origin) back to the emergence of the content (Sect. 10.1) is automatically encoded into the sign and that (ii) for successful interpretation this perspective must be reconstructucted from the sign by the hearer. DBS formalizes the speaker's perspective as from the STAR-1 to the STAR-0, and the hearer's reconstruction of this perspective from the interpretation situation's STAR-2 to the STAR-1. For meaningful production and correct interpretation, knowing a language sign's STAR-1 is essential (Sect. 10.2).

In face to face communication, the hearer gets to know the sign's origin as an eye- and/or earwitness. In recorded, e.g. written, language, in contrast, the hearer's situation of interpretation, represented formally by the STAR-2, may be arbitarily far removed from the sign's origin. To be understood, it is therefore in the interest of the writer to specify the utterance situation directly in the recorded sign (e.g. a letter or a text). Otherwise, the reader will have difficulty anchoring the written sign to its context of utterance, which would compromise a correct and complete interpretation.

As an example, consider the interpretation of the following text sample:

11.1.1 Text with a Dispersed Coding of the STAR-1

> Jan. 16th, 1832 – The neighbourhood of Porto Praya, viewed from the sea, wears a desolate aspect. The volcanic fire of past ages, and the scorching heat of the tropical sun, have in most places rendered the soil steril and unfit for vegetation. The country rises in successive steps of table land, interspersed with some truncate conical hills, and the horizon is bounded by an irregular chain of more lofty mountains. The scene, as beheld through the hazy atmosphere of this climate, is one of great interest; if, indeed, a person, fresh from the sea, and who has just walked, for the first time, in a grove of cocoa-nut trees, can be a judge of anything but his own happiness.

Charles Darwin 1839, *Voyage of the Beagle*, p. 41

How is it possible for the speaker/writer to code a three-dimensional landscape into a one-dimensional sequence of word form surfaces which the hearer/reader can decode into an image of that same three-dimensional landscape in his or her mind? And not only is a landscape being described by this string of word forms, but also the author's feelings of going on land after a long voyage in the confined quarters of the HMS Beagle.[1]

In DBS, the answer has three parts. The first is the implementation of the agent's basic recognition and action procedures and reusing them as the core values of proplets (grounding); attached to English surfaces by convention, proplets double as language meanings. The second part is the concatenation of the proplets into content by means of classic functor-argument and coordination, encoded by the morphological properties of the English surfaces and their time-linear order. The third part is the anchoring of the text's content to its STAR-1, which specifies the origin of the content's surface regarding Space, Time, Author, and intended Recipient (addressee).

For the reader, this raises the question of how to find the values of a written sign's STAR-1. For a human, finding these values in a book is usually easy, even if they are dispersed in various places, as in the above example: The S value Porto Praya of the appropriate STAR-1 is embedded into the first sentence of the text. The T value 1832-01-16 is given as the diary date Jan. 16th, 1832 preceding the text. The A value equals the author, as stated on the book's cover. And the R value is the general readership.

Other possible choices are the S value London and the T value 1839 for where and when, respectively, the book was first published. These may be appropriate for other texts, but not for our example. Such ambiguities between appropriate and inappropriate STAR-1 values present a difficulty for an artificial agent. They may be resolved by standardizing the specification of the STAR-1 for newly written texts and by defining templates for different kinds of books by different authors for existing texts – plus human help when needed.

[1] British Cherokee class (ten-gun brig) sailing ship designed in 1807 and built in 1818–1820.

Often the author's circumstances of utterance, coded in the STAR-1, differ from the STAR-0 (Sect. 10.2); the STAR-0 codes the circumstances of a content's emergence as a monitoring and recording of the agent's current recognition and action (Sect. 10.1), or of current reasoning (inferencing). However, if the agent reports the recorded data immediately, in speech or writing, the STAR-1 and the STAR-0 are practically the same. In this case, the perspective of the STAR-1 is shortened to the perspective of the STAR-0.

Mapping a STAR-0 = STAR-1 content into language creates an effect of immediacy and authenticity. This is what the author achieves in example 11.1.1: there is no direct reference to any earlier events or locations of a separate STAR-0 content. Instead, the use of the present tense leads the reader to view Porto Praya with the eyes of the author from the deck of the Beagle on Jan. 16th, 1832. The reader can relax and enjoy the author's report because he or she is neither expected nor even able to derive a response which could reach the author, in contradistinction to a face-to-face communication.

In DBS, a text like 11.1.1 is analyzed as a sequence of statement dialogs. As shown in 10.6.7, each elementary statement dialog includes the derivation of a STAR-2 perspective by the hearer/reader (otherwise it would be a monologue). In 11.1.1, the STAR-2 perspective is not used as the basis for a subsequent hearer/reader response (in contradistinction to Sect. 10.3). It is not even used for interpreting indexicals like pro1 and pro2 (because there are none in the text). It only provides a perspective as an integral part of the reader's understanding, looking back from her or his current STAR-2 circumstances to the author's STAR-0 = STAR-1 circumstances in the year 1832.

While the computation of the speaker/writer and the hearer/reader perspectives based on the STAR-0, STAR-1, and STAR-2 is completely software-mechanical, there is another aspect of understanding which relates to differences in the background knowledge of different readers (FoCL Sect. 21.5). For example, a reader who knows the location of Porto Praya on the globe will have a better understanding of the sample text than one who does not. Similarly, a reader who has personally experienced a walk under cocoa-nut trees or has traveled for a month on a sailing ship will have a more complete understanding of the text than one who has not.

Differences in individual background knowledge (FoCL Sect. 5.3) apply to the understanding not only of statements, but also of questions and requests. As unsystematic variations, they may be easily modeled by providing the word banks of different artificial agents with different contents. In short, background variation is neither an intrinsic part of nor an obstacle to reconstructing the language communication mechanism as a software procedure.

11.2 Indexical vs. Coreferential Use of 3rd Person Pronouns

Ambiguity in natural language is limited to the hear mode. This applies also to the alternative between an indexical and a coreferential interpretation of 3rd person pronouns (HBTR Sects. 3.3, 3.5). Consider the following example:

11.2.1 AMBIGUOUS HEAR MODE INTERPRETATION OF she

After *Mary* came home, *she* took a bath.

Here, the personal pronoun *she* may be either (i) coreferent with *Mary* or (ii) refer indexically to another female present in the context of use, e.g. *Suzy*. While the speaker knows what he or she wants to say, the hearer has to choose between two substantially different interpretations.

1st and 2nd person pronouns differ from 3rd person pronouns in that the coreferential interpretation of the former is restricted to direct (quoted) speech:

11.2.2 COREFERENTIAL INTERPRETATION OF 1ST AND 2ND PERSON

John told Mary: *I* love *you*. Mary told John: *I* love *you*.$^{\text{STAR}-1}$

Here, the pronouns *I* and *you* can not be interpreted relative to the A and R values of the STAR-1, in contradistinction to the statement dialog in Sect. 10.3. Instead, they are interpreted as coreferential with the subject and the object of the sentence into which they are embedded.

Outside direct speech, 1st and 2nd person pronouns are restricted to an indexical interpretation:

11.2.3 INDEXICAL INTERPRETATION OF 1ST AND 2ND PERSON

I told *you* that *I* love *you* and *you* told *me* that *you* love *me*.$^{\text{STAR}-1}$

Here, the 1st and 2nd person pronouns point at the A and the R value, respectively, of the STAR-1.

The formal analysis of 3rd person pronouns must distinguish between their representation in (i) the lexicon, (ii) the speak mode, and (iii) the hear mode. The lexical representation uses pro3 as the core value, which is neutral regarding indexical vs. coreferential use. If the indexical interpretation is selected in the speak mode, pro3 is used; if the coreferential interpretation is selected, the address of the referent, e.g. (Mary 23), is used as the core value. If the choice is open (the pragmatic ambiguity, FoCL 12.3.5) both representations connected by a slash, e.g. pro3/(Mary 23), are used (11.3.2).

In addition to the core values, personal pronouns are lexically differentiated by grammatical properties, as in the following proplet definitions:

11.2.4 Lexical analysis of personal pronouns in English

sur: I	sur: me	sur: you	sur: he	sur: him	sur: she
noun: pro1	noun: pro1	noun: pro2	noun: pro3	noun: pro3	noun: pro3
cat: s1	cat: obq	cat: sp2	cat: s3	cat: obq	cat: s3
sem: sg	sem: sg	sem:	sem: m sg	sem: m sg	sem: f sg
...	...	...	...	...	...
prn:	prn:	prn:	prn:	prn:	prn:

sur: her	sur: it	sur: we	sur: us	sur: they	sur: them
noun: pro3	noun: pro3	noun: pro1	noun: pro1	noun: pro3	noun: pro3
cat: obq	cat: snp	cat: p1	cat: obq	cat: p3	cat: obq
sem: f sg	sem: -mf sg	sem: pl	sem: pl	sem: pl	sem: pl
...	...	...	...	...	...
prn:	prn:	prn:	prn:	prn:	prn:

The sem values sg (singular), pl (plural), m (masculine), and f (feminine) are needed for defining agreement with possible coreferents. The cat values s1, sp2, s3, p1, and p3 (nominatives specified for number and person) and obq (oblique or non-nominative) are needed for defining agreement with the finite verb.

The hearer's choice between the indexical and the coreferential interpretation[2] of a 3rd person pronoun is not a matter of different lexical pro3 entries, but of different uses (pragmatics). This may be shown graphically (FoCL Sect. 6.3) with the following example of a clausal object construction (11.3.4):

11.2.5 Pragmatically ambiguous hear mode content

Mary knew that *she* was happy.$^{\text{STAR}-2}$

One interpretation is the indexical use: instead of *she* being coreferential with *Mary*, pro3 is interpreted as pointing at another female, e.g. *Suzy*:

11.2.6 Indexical use of she

Mary knew that she was happy.

On the coreferential use, in contrast, pro3 is interpreted as being coreferential with *Mary*.

11.2.7 Coreferential use of she

Mary knew that she was happy.

[2] Bosch (1986) uses the notions "syntactic reference" for our coreferential reference and "referential reference" for our indexical reference.

In linguistics, the "full noun"[3] referred to by a coreferential pronoun is called the *antecedent* if it precedes, and the *postcedent* if it follows the pronoun.

The pragmatic ambiguity between the indexical and the coreferential use arises in STAR-2 contents after parsing the incoming surface and before adjusting to the hearer perspective (10.3.1). No such ambiguity arises in a corresponding STAR-0 or STAR-1 content because the agent producing the STAR-1 content chooses whether the intended referent should be represented indexically or coreferentially, without or with language.

An indexical interpretation of 11.2.5 is illustrated by the following content, with pro3 pointing at the value of the 3rd attribute in the co-indexed STAR-0:

11.2.8 Indexical STAR-0 representation as a proplet set

$$
\begin{bmatrix} \text{noun: Mary} \\ \text{cat: snp} \\ \text{sem: nm f sg} \\ \text{fnc: know} \\ \text{prn: 89} \end{bmatrix}
\begin{bmatrix} \text{verb: know} \\ \text{cat: \#n}' \text{ \#a}' \text{ decl} \\ \text{sem: past} \\ \text{arg: Mary (happy 90)} \\ \text{prn: 89} \end{bmatrix}
\begin{bmatrix} \text{noun: pro3} \\ \text{cat: s3} \\ \text{sem: sg f} \\ \text{fnc: happy} \\ \text{prn: 90} \end{bmatrix}
\begin{bmatrix} \text{verb: happy} \\ \text{cat: \#ns13}' \text{ \#be}' \text{v} \\ \text{sem: } \textit{that} \text{ past} \\ \text{arg: pro3} \\ \text{fnc: (know 89)} \\ \text{prn: 90} \end{bmatrix}
\begin{bmatrix} \text{S: Austin} \\ \text{T: 1974-09-12} \\ \text{A: Peter} \\ \text{3rd: Suzy} \\ \text{prn: 89–90}^{4} \end{bmatrix}
$$

The agent, Peter, monitors his current environment and anchors the content to the current STAR-0 proplet. If the monitoring includes a reference to *Suzy*, it may be by the contextual counterparts of any of the three reference mechanisms of natural language, i.e. by symbol, by pointer, or by baptism (HBTR Sects. 3.2–3.4). If the pointer is chosen, the STAR-1 perspective for a language production will realize it as the pronoun she.

If Peter's STAR-0 monitoring includes two references to *Mary* instead, one of them may be by coreference, using the address (Mary 93). When he computes the STAR-1 perspective, this address may also be realized as a pronoun.

11.2.9 Coreferential STAR-0 representation as a proplet set

$$
\begin{bmatrix} \text{noun: Mary} \\ \text{cat: snp} \\ \text{sem: nm f sg} \\ \text{fnc: know} \\ \text{prn: 93} \end{bmatrix}
\begin{bmatrix} \text{verb: know} \\ \text{cat: \#n}' \text{ \#a}' \text{ decl} \\ \text{sem: past} \\ \text{arg: Mary (happy 94)} \\ \text{prn: 93} \end{bmatrix}
\begin{bmatrix} \text{noun: (Mary 93)} \\ \text{cat: s3} \\ \text{sem: sg f} \\ \text{fnc: happy} \\ \text{prn: 94} \end{bmatrix}
\begin{bmatrix} \text{verb: happy} \\ \text{cat: \#ns13}' \text{ \#be}' \text{ v} \\ \text{sem: } \textit{that} \text{ past} \\ \text{arg: (Mary 93)} \\ \text{fnc: (know 93)} \\ \text{prn: 94} \end{bmatrix}
\begin{bmatrix} \text{S: Austin} \\ \text{T: 1974-09-12} \\ \text{A: Peter} \\ \text{prn: 93} \end{bmatrix}
$$

Because this content does not contain any pro3 pointer, the STAR-0 needs no 3rd feature for its interpretation.

In summary, a pronoun allowing different interpretations in a STAR-1 surface creates a pragmatic ambiguity for the hearer. For computing the STAR-2

[3] Or rather the *referent* of the full noun.

[4] To facilitate the pointing of pro3 to the value of the 3rd attribute the prn value of the STAR-0 may be set to an interval, here 89–90.

perspective, the hearer must make a choice and try to select the interpretation intended by the speaker. There are three possibilities: selecting (i) the same interpretation as the speaker (correct), (ii) a different interpretation (incorrect), or (iii) remaining undecided (inconclusive). In the latter case there may be the option to request clarification from the speaker.[5]

11.3 Langacker-Ross Constraint for Clausal Arguments

It has been observed by Langacker (1969) and Ross (1969) that 3rd person pronouns in certain subclause constellations can *only* have an indexical interpretation. This constraint, known as the Langacker-Ross constraint, applies to all the kinds of subclause construction in natural language. These are (i) the clausal subject (NLC Sect. 7.1), (ii) the clausal object (NLC Sect. 7.2), (iii) the clausal adnominal modifier (aka relative clause, NLC Sects. 7.3, 7.4), and (iv) the clausal adverbial modifier (NLC Sect. 7.5).

Consider the following variants of a clausal subject construction:

11.3.1 PRONOUN IN CLAUSAL SUBJECT CONSTRUCTIONS

1. LH' Coreferent noun in clause lower (L) precedes pronoun in higher clause (H'): That *Mary* was happy surprised *her*.
2. H'L Pronoun in higher clause (H') precedes non-coreferential noun in lower clause (L): % *She* was surprised that *Mary* was happy.
3. L'H Pronoun in lower clause (L') precedes coreferent noun in higher clause (H): That *she* was happy surprised *Mary*.
4. HL' Coreferent noun in higher clause (H) precedes pronoun in lower clause (L'): *Mary* was surprised that *she* was happy.

All four sentences are well-formed. They differ as to (i) whether the lower clause L (subclause) precedes (LH) or follows (HL) the higher clause H, and (ii) whether the pronoun is in the lower clause (L') or in the higher clause (H'). The constructions 1, 3, and 4 are alike in that they permit an indexical as well as a coreferential interpretation. Construction 2, in contrast, allows only the indexical interpretation (11.2.6); this restriction (compared to the three other examples) is indicated by the % marker.

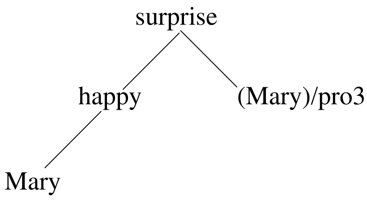
[5] By treating the pragmatic ambiguity between an indexical and a coreferential interpretation erroneously as syntactic and combining it with a syntactic recursion (8.3.1), a recursive ambiguity may be constructed. Even if the mistaken treatment as a syntactic ambiguity were accepted for the sake of the argument, the recursive ambiguity may (and should) be avoided by using semantic doubling (FoCL Sect. 12.5).

Given that the choice of a pronoun and its form of reference (indexical vs. coreferential) originate in the speaker, let us consider the DBS graph analyses of these constructions. We begin with the first two examples of 11.3.1, which have the pronoun in the higher clause. They share the SRG, the signature, and the NAG, but have two different surface realizations, (a) and (b):

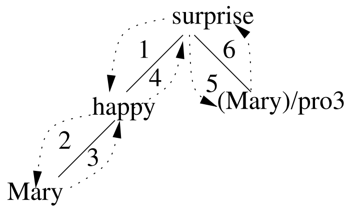
11.3.2 Clausal subject: pronoun in higher clause (matrix)

1. **LH':** That *Mary* was happy surprised *her*.
2. **H'L:** % *She* was surprised that *Mary* was happy.

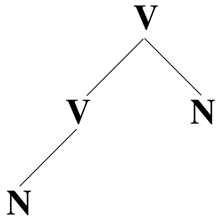
(i) *SRG (semantic relations graph)*



(iii) *NAG (numbered arcs graph)*



(ii) *signature*



(iv) *surface realization*

a.

1	2	3	4	5	6
That	Mary	was_happy	surprised	her	•
V/V	V/N	N/V	V/V	N\V	N/V

b.

5	6	1	2	3	4
% She	was_surprised	that	Mary	was_happy	•
V\N	N\V	V/V	V/N	N/V	V/V

For ease of exposition, the node allowing a coreferential as well as an indexical interpretation is represented as (Mary)/pro3. The address (Mary) stands for the speaker's choice of a coreferential reference (as in 11.2.9), while pro3 stands for the choice of an indexical reference (as in 11.2.8).

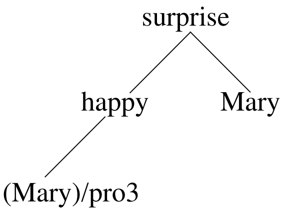
The surface realization (a) allows an indexical as well as a coreferential interpretation of her. The surface realization (b), in contrast, allows only an indexical interpretation of she, as indicated by the %-marker.

Next consider the DBS graph analysis of examples 3 and 4 in 11.3.1. As in 11.3.2, they have the same (i) SRG, (ii) signature, and (iii) NAG, but two different (iv) surface realizations.

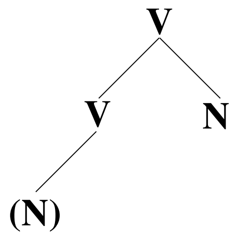
11.3.3 Clausal subject: pronoun in lower clause

3. **L'H:** That *she* was happy surprised *Mary*.
4. **HL':** *Mary* was surprised that *she* was happy.

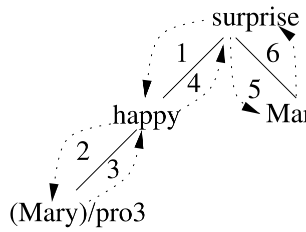
(i) *SRG (semantic relations graph)*



(ii) *signature*



(iii) *NAG (numbered arcs graphs)*



(iv) *surface realization*

a.	1	2	3	4	5	6
	That	she	was_happy	surprised	Mary	.
	V/V	V/N	N/V	V/V	V\N	N\V

b.	5	6	1	2	3	4
	Mary	was_surprised	that	she	was_happy	.
	V\N	N\V	V/V	V/N	N/V	V/V

In contradistinction to 11.3.2, the (Mary)/pro3 node in the SRG and in the NAG is positioned in the lower clause. The surface realizations (a) and (b) are unrestricted in that each has an indexical and a coreferential interpretation. In summary, the variants 1, 3, and 4 in 11.3.1 are pragmatically equivalent in that they allow the indexical as well as the coreferential interpretation, while variant 2 is unambiguous in that it shares only the indexical interpretation.

The other kind of sentential argument is the clausal object. Let us construct the LH', H'L, L'H, and HL' constellations of a 3rd person pronoun and its potential antecedent similar to 11.3.1:

11.3.4 Pronoun in clausal object constructions

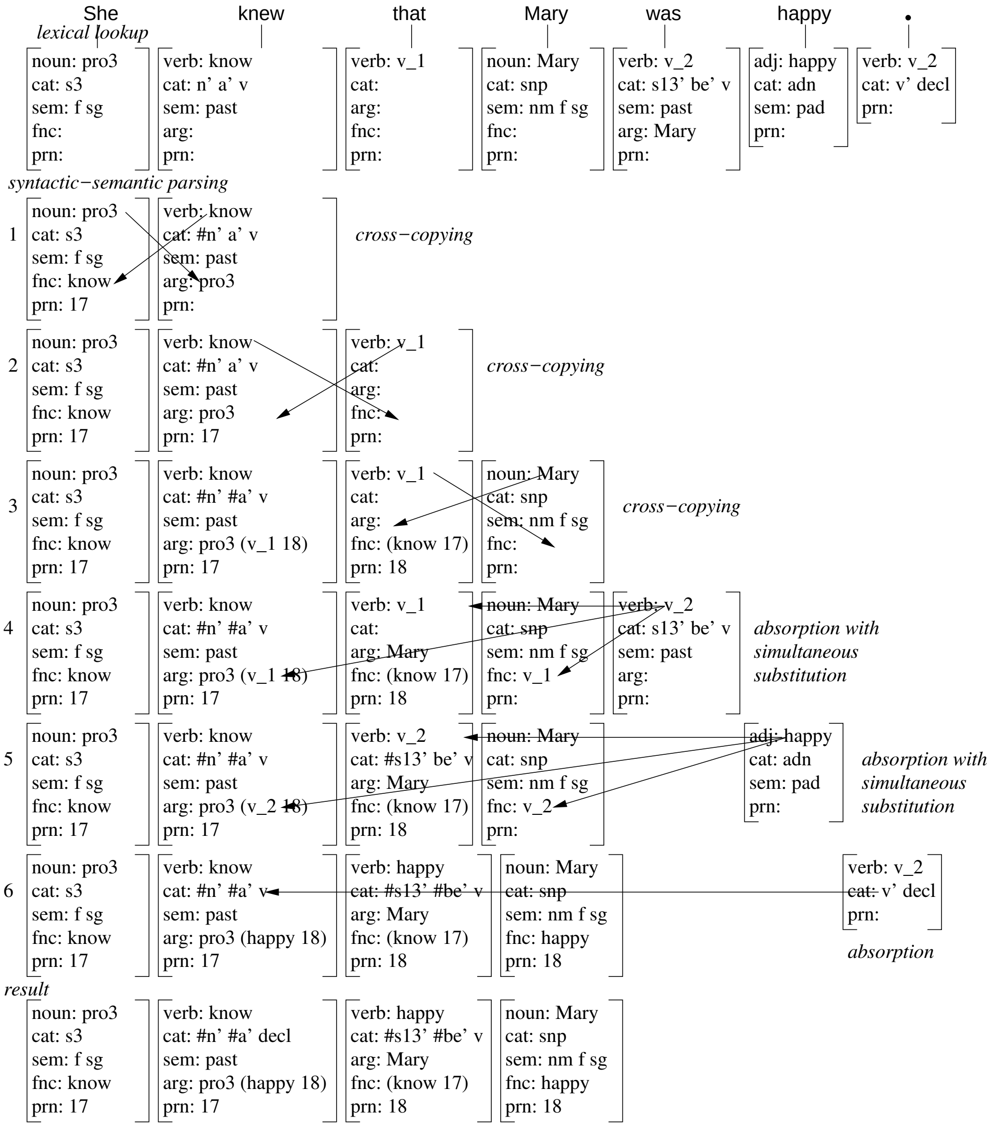
1. LH' Coreferent noun in lower clause (L) precedes pronoun in higher clause (H'): That *Mary* was happy was known to *her*.
2. H'L Pronoun in higher clause (H') precedes non-coreferential noun in lower clause (L): % *She* knew that *Mary* was happy.
3. L'H Pronoun in lower clause (L') precedes coreferent noun in higher clause (H): That *she* was happy was known to *Mary*.
4. HL' Coreferent noun in higher clause (H) precedes pronoun in lower clause (L'): *Mary* knew that *she* was happy.

Again, the alternative ordering of the higher and the lower clause is based on using passive (9.2.2) instead of active (9.2.1) in the higher clause of sentences

1 and 3. Languages with a word order freer than English may use fronting of the clausal object as an alternative to a passive construction.[6]

Consider the hear mode derivation of the H'L construction of example 2 in 11.3.4, which allows only an indexical interpretation of the pronoun she.

11.3.5 Hear mode derivation of H'L construction (example 2)



The hear mode readings of examples 1 and 2 in 11.3.4 are as follows:

---

[6] For example, sentence 1 in 11.3.4 may be translated into German as Dass sich Maria freute, wurde von ihr bemerkt. (passive) or Dass sich Maria freute, bemerkte sie. (fronting of the clausal object using free word order).

11.3.6 LH’:That she was happy was known to Mary (example 1)

noun: (Mary)/pro3	verb: happy	noun: Mary	verb: know
cat: s3	cat: #ns13′ #be′ v	cat: snp	cat: #n′ #a′ decl
sem: f sg	sem: *that* past	sem: nm f sg	sem: past
fnc: happy	arg: (Mary)/pro3	fnc: know	arg: Mary (happy 17)
mdr:	fnc: (know 18)	mdr:	mdr:
prn: 17	prn: 17	prn: 18	prn: 18

11.3.7 H’L: % She knew that Mary was happy (example 2)

noun: pro3	verb: know	noun: Mary	verb: happy
cat: s3	cat: #n′ #a′ decl	cat: snp	cat: #ns13′ #be′ v
sem: f sg	sem: past	sem: nm f sg	sem: *that* past
fnc: know	arg: pro3 (happy 18)	fnc: happy	arg: Mary
mdr:	mdr:	mdr:	fnc: (know 17)
prn: 17	prn: 17	prn: 18	prn: 18

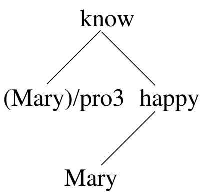
The alternative interpretations in 11.3.6 are coded by the value (Mary)/pro3. The restriction to indexical use in 11.3.7, in contrast, is indicated by the value pro3 instead.

For the speak mode, consider the corresponding DBS graph analyses:

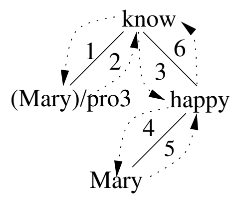
11.3.8 CLAUSAL OBJECT: PRONOUN IN HIGHER CLAUSE

1. **LH’:** That *Mary* was happy was known to *her*.
2. **H’L:** % *She* knew that *Mary* was happy.

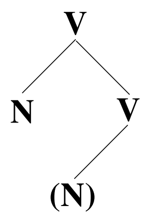
(i) *SRG (semantic relations graph)*



(iii) *NAG (numbered arcs graph)*



(ii) *signature*



(iv) *surface realization*

a.

3	4	5	6	1	2
That	Mary	was_happy	was_known	to_her	.
V\V	V/N	N/V	V\V	V/N	N/V

b.

	1	2	3	4	5	6
%	She	knew	that	Mary	was_happy	.
	V/N	N\V	V\V	V/N	N/V	V\V

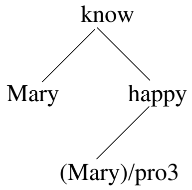
In the SRG and the NAG, the subject of the higher clause is represented by the (Mary)/pro3 node. In the surface realization (a) this node has an indexical and a coreferential interpretation. The surface realization (b), in contrast, has an indexical interpretation only, as indicated by the % marker. This is similar to the asymmetry between the clausal subject constructions 1 and 2 in 11.3.1.

Next consider the DBS graph analysis of constructions 3 and 4 listed in 11.3.4, with the pronoun in the lower clause:

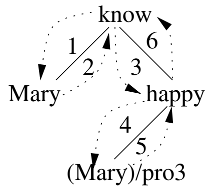
11.3.9 CLAUSAL OBJECT: PRONOUN IN LOWER CLAUSE

3. **L'H:** That *she* was happy was known to *Mary*.
4. **HL':** *Mary* knew that *she* was happy.

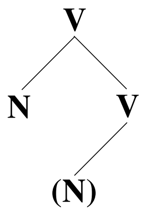
(i) *SRG (semantic relations graph)*



(iii) *NAG (numbered arcs graph)*



(ii) *signature*



(iv) *surface realization*

a.

3	4	5	6	1	2
That	she	was_happy	was_known	to_Mary	.
V\V	V/N	N/V	V\V	V/N	N/V

b.

1	2	3	4	5	6
Mary	knew	that	she	was_happy	.
V/N	N/V	V\V	V/N	N/V	V\V

The surface realizations (a) and (b) are based on alternative traversals of the NAG. They are unrestricted in that each allows an indexical as well as a coreferential interpretation (similar to 11.3.3).

11.4 Coreference in Clausal Adnominal Modifiers

In English, clausal adnominal modifiers (aka relative clauses) are positioned directly behind their modified ("head noun") – unless they are extraposed (9.3.5). Therefore, the subclause may be located in the middle of the main clause (cf. 1 and 3 in 11.4.1), unlike clausal subjects and objects. However, because the restriction observed by Langacker and Ross applies to the order of the pronoun and the coreferent noun (and not the order of the subclause and the main clause), the constraint applies in full to clausal adnominals:[7]

11.4.1 PRONOUN IN ADNOMINAL MODIFIER CONSTRUCTIONS

1. LH' Coreferent noun in lower clause (L) precedes pronoun in higher clause (H'): The man who loves *the woman* kissed *her*.
2. H'L Pronoun in higher clause (H') precedes non-coreferential noun in lower clause (L): % *She* was kissed by the man who loves *the woman*.

3. L'H Pronoun in lower clause (L') precedes coreferent noun in higher clause (H): The man who loves *her* kissed *the woman*.
4. HL' Coreferent noun in higher clause (H) precedes pronoun in lower clause (L'): *The woman* was kissed by the man who loves *her*.

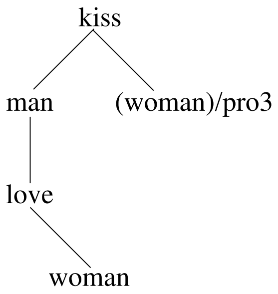
The four constructions are analogous to those of 11.3.1 and 11.3.4. Again, it is the H'L constellation which is limited to indexical reference.

The first two examples have the following DBS graph analysis:

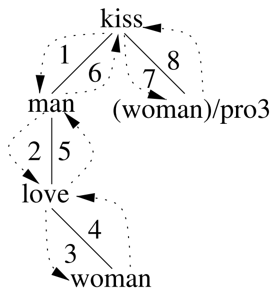
11.4.2 Clausal adnominal modifier: pronoun in higher clause

1. **LH':** The man who loves *the woman* kissed *her*.
2. **H'L:** % *She* was kissed by the man who loves *the woman*.

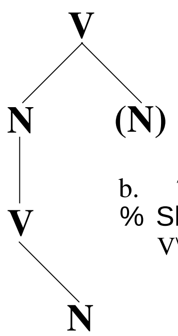
(i) *SRG (semantic relations graph)*



(iii) *NAG (numbered arcs graph)*



(ii) *signature*



(iv) *surface realization*

a.

1	2	3	4	5	6	7	8
The_man	who_loves	the_woman			kissed	her	.
V/N	N|V	V\N	N\V	V|N	N/V	V\N	N\V

b.

7	8	1	2	3	4	5	6
% She	was_kissed	by_the_man	who_loves	the_woman			.
V\N	N\V	V/N	N|V	V/N	N\V	V|N	N\V

Similar to the analyses of clausal arguments (Sect. 11.3), the ambiguous node in the NAG is represented as (woman)/pro3. In the surface realization (a) this node has an indexical and a coreferential interpretation. The surface realization (b), in contrast, has an indexical interpretation only, as indicated by the % marker. This is similar to the clausal subject (11.3.2) and object (11.3.8) constructions.

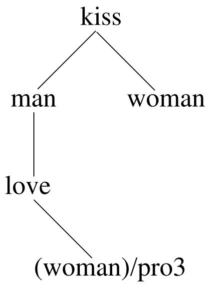
[7] For a more detailed discussion of clausal adnominal modifier constructions (relative clauses) apart from the interpretation of pronouns see NLC Sects. 7.3 and 7.4; TExer Sects. 3.3. and 3.4.

The following constructions with the pronoun in the lower clause, in contrast, are ambiguous between an indexical and a coreferential interpretation.

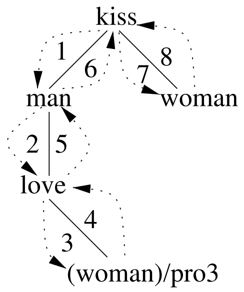
11.4.3 CLAUSAL ADNOMINAL MODIFIER: PRONOUN IN LOWER CLAUSE

3. **L'H:** The man who loves *her* kissed *the woman*.
4. **HL':** *The woman* was kissed by the man who loves *her*.

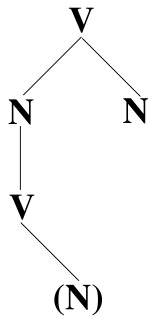
(i) *SRG (semantic relations graph)*



(iii) *NAG (numbered arcs graph)*



(ii) *signature*



(iv) *surface realization*

a.

1	2	3	4	5	6	7	8
The_man	who_loves	her			kissed	the_woman	•
V/N	N|V	V/N	N\V	V|N	N/V	V\N	N\V

b.

7	8	1	2	3	4	5	6
The_woman	was_kissed	by_the_man	who_loves	her			•
V\N	N\V	V/N	N|V	V\N	N\V	V|N	N/V

The pragmatically ambiguous node in the SRG and the NAG is (woman)/pro3.

The LH' constellation (1) in 11.4.1 is the cause of an empirical difficulty for truth-conditional semantics. First observed by Geach (1962), his example is known as the "donkey sentence":

11.4.4 THE DONKEY SENTENCE

Every farmer who owns a donkey beats it.

To the untutored, the seemingly most natural translation of the coreferential interpretation of the pronoun it into truth-conditional semantics (i.e. predicate calculus used for the analysis of natural language meaning) is the following:

11.4.5 INADEQUATE QUANTIFIER STRUCTURE FOR DONKEY SENTENCE

$\forall x\ [[\text{farmer}(x) \wedge \exists y\ [\text{donkey}(y) \wedge \text{own}(x,y)] \rightarrow \text{beat}(x,y)]$

Unfortunately, the y in beat(x,y) is not in the scope of the quantifier $\exists$y binding donkey(y) in the subordinate clause, as pointed out by Geach (op. cit.).[8]

The DBS solution is based on disentangling three different tasks which predicate calculus ties in a knot.

11.4.6 Tasks which predicate calculus syntax ties in a knot

1. In predicate calculus, the quantifiers $\forall$x and $\exists$y serve as the rudimentary determiners *every* and *some*, respectively. In DBS, the determiner task is replaced by the values exh(austive), sel(ective), def(inite), indef(inite), sg, and pl, used in the sem attribute of noun proplets.[9]
2. In predicate calculus, functors like donkey and own are related by using the same variable as arguments, as in donkey(y) and own(x, y), and binding the common variable with the quantifiers $\forall$y or $\exists$y. In DBS, the binding task is replaced by address values coding interproplet relations.
3. In predicate calculus, elementary propositions like farmer(x) and donkey(y) are combined using the connectives and the coordinating syntax of propositional calculus (6.4.1). In DBS, the combination task is replaced by a common prn value holding the proplets of a proposition together.

In short, instead of restructuring the content of 11.4.4 beyond recognition (and without success), as in 11.4.5, the DBS alternative is a standard, automatic, time-linear, surface compositional hear mode derivation analogous to the adnominal (relative) clause example in NLC 7.3.2. The quantifiers binding variables horizontally, causing the scope problem in 11.4.5, are set aside and replaced by address values which establish coreference between it and donkey successfully. The content of 11.4.4 is coded as the following set of proplets:

11.4.7 Representing the donkey content as a set of proplets

noun: farmer	verb: own	noun: donkey	verb: beat	noun: (donkey 17)
cat: snp	cat: #n′ #a′ v	cat: snp	cat: #ns3′ #a′ decl	cat: snp
sem: pl exh	sem: pres	sem: indef sg	sem: pres	sem: sg
fnc: beat	arg: $\emptyset$ donkey	fnc: own	arg: farmer (donkey 17)	fnc: beat
mdr: (own 17)	mdd: (farmer 16)	mdr:	mdr:	mdr:
prn: 16	prn: 17	prn: 17	prn: 16	prn: 16

The noun proplet *farmer* has the sem values pl exh (plural exhaustive) in combination with the cat value snp (singular noun phrase), characterizing the

[8] Discourse representation theory (Kamp 1981, Kamp and Reyle 1993, Geurts 2002) originated as an attempt to resolve this well-known problem for truth-conditional semantics of natural language.

[9] See NLC 6.4.7 for a set-theoretic characterization of these values.

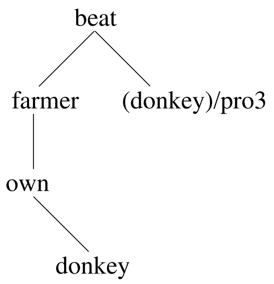
determiner every. The noun proplet *donkey* has the cat value snp in combination with the sem values indef sg (indefinite singular), characterizing the determiner a(n). The coreference of it with the antecedent donkey is coded in the rightmost proplet with the address core value (donkey 17).

The main clause and the adnominal subclause have different prn values, here 16 and 17. The modifier|modified relation between the subclause verb *own* and the main clause noun *farmer* is coded by their mdd and mdr features with the address values (farmer 16) and (own 17), respectively. The subject gap of the adnominal subclause is indicated by ∅ in the initial arg slot of *own*.

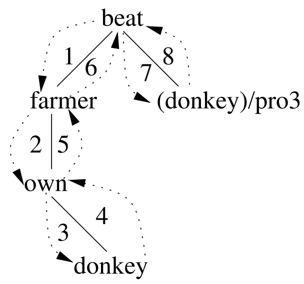
The content 11.4.7 has the following DBS graph analysis:

11.4.8 DBS GRAPH ANALYSIS OF THE DONKEY SENTENCE (LH')

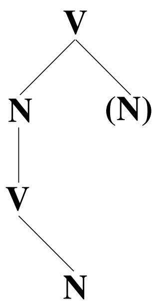
(i) *SRG (semantic relations graph)*



(iii) *NAG (numbered arcs graph)*



(ii) *signature*



(iv) *surface realization*

1	2	3	4	5	6	7	8
Every_farmer	who_owns	a_donkey			beats	it	.
V/N	N|V	V\N	N\V	V|N	N/V	V\N	N\V

The pragmatic ambiguity between an indexical and a coreferential interpretation of the pronoun it is represented by the value (donkey)/pro3.

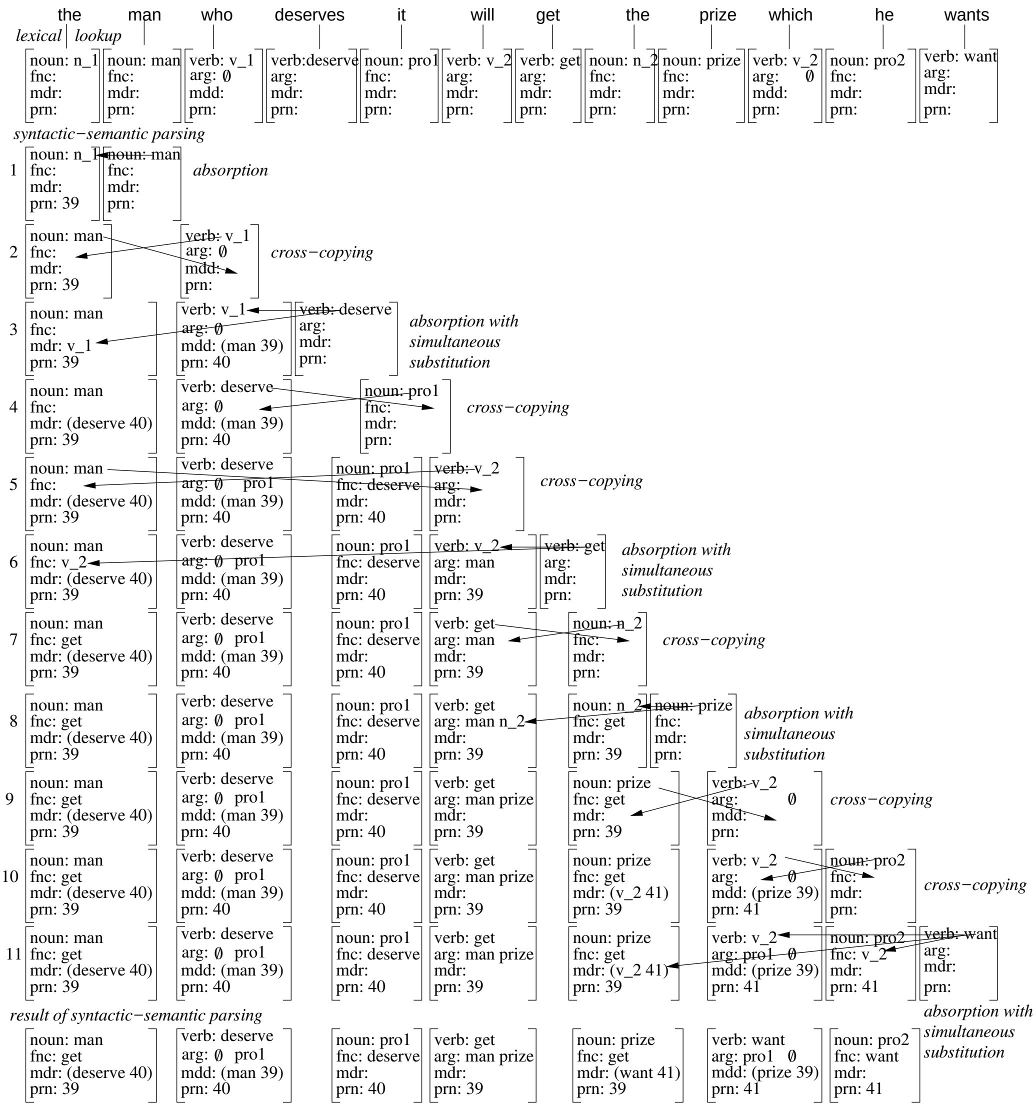
Clausal adnominal modifiers cause an empirical difficulty not only for truth-conditional semantics, but also for nativist syntax. It is caused by intertwined L'H and HL' relative clauses, one with a subject gap (NLC Sect. 7.3), the other with an object gap (NLC Sect. 7.4), and known as the "Bach-Peters sentence."

11.4.9 THE BACH-PETERS SENTENCE

The man who deserves it will get the prize which he wants.

The DBS hear mode derivation is completely standard:

11.4.10 DBS HEAR MODE DERIVATION OF BACH-PETERS SENTENCE



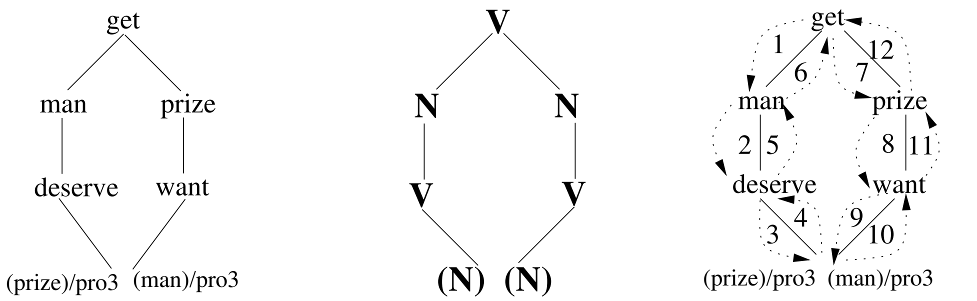
In transformational grammar, this construction caused a problem because the pronoun it was derived transformationally from the "underlying" noun phrase the prize which he wants, and the pronoun he from the man who deserves it. Because each underlying noun phrase contains a pronoun which itself is based on an underlying noun phrase there result two recursions leading to an infinite "deep structure." Based on this analysis, Peters and

Ritchie (1973) proved that transformational grammar has the complexity degree ‘undecidable’ (which is computationally intractable).

The alternative analysis of this example in DBS is as efficient as it is simple, like that of the center-embedded relative clauses shown in 9.3.2:

11.4.11 DBS GRAPH ANALYSIS OF THE BACH-PETERS SENTENCE

(i) *SRG (sem. rel. graph)* (ii) *signature* (iii) *NAG (num. arcs graph)*



(iv) *surface realization*

1	2	3	4	5	6	7	8	9	10	11	12
The_man	who_deserves	it			will_get	the_prize	which	he	wants		.
V/N	N|V	V\N	N\V	N|V	N/V	V\N	N|V	V/N	N/V	V\N	N\V

The graph indicates that the first relative clause who deserves it has a subject gap, while the second relative clause (which)[10] he wants has an object gap.

The representation of the content as a set of proplets is as follows:

11.4.12 CONTENT REPRESENTATION OF THE BACH-PETERS SENTENCE

noun: man	verb: deserve	noun: (prize 39)
cat: snp	cat: #ns3′ #a′ v	cat: snp
sem: def sg	sem: *who* ind pres	sem: sg -mf
fnc: get	arg: ∅ (prize 39)	fnc: deserve
mdr: (deserve 40)	mdd: (man 39)	mdr:
prn: 39	prn: 40	prn: 40

verb: get
cat: #n′ #a′ decl
sem: fut
arg: man prize
mdr:
prn: 39

noun: prize	noun: (man 39)	verb: want
cat: snp	cat: s3	cat: #ns3′ #a′ v
sem: def sg	sem: sg m	sem: *which* pres
fnc: get	fnc: want	arg: (man 39) ∅
mdr: (want 41)	mdr:	mdd: (prize 39)
prn: 39	prn: 41	prn: 41

The pronoun it is represented by the noun proplet with the address (prize 39) as its core value; the pronoun he is represented by the noun proplet with the address (man 39) as its core value (two coreferential interpretations). The implicit fillers of the arg values ∅ in the *deserve* and *want* proplets are specified by the respective mdd features directly underneath.

The extrapropositional semantic relation between the first noun proplet *man* of the main clause and the first relative clause is coded by the modifier value (deserve 40) in the mdr attribute of *man* and the modified value (man 39) in the mdd attribute of *deserve*. The extrapropositional semantic relation between the second noun proplet *prize* of the main clause and the second relative clause is coded by the modifier value (want 41) in the mdr attribute of *prize* and the modified value (prize 39) in the mdd attribute of *want*.

11.5 Coreference in Clausal Adverbial Modifiers

The four remaining subclause constructions subject to the Langacker-Ross constraint are clausal adverbial modifiers.[11] The following constellations correspond to those of 11.3.1 (clausal subject), 11.3.4 (clausal object), and 11.4.1 (clausal adnominal modifier):

11.5.1 LANGACKER-ROSS CONSTRAINT IN ADVERBIAL SUBCLAUSES

1. LH' Coreferent noun in lower clause (L) precedes pronoun in higher clause (H'): When *Mary* returned *she* kissed John.

2. H'L Pronoun in higher clause (H') precedes non-corefential noun in lower clause (L): % *She* kissed John when *Mary* returned.

3. L'H Pronoun in lower clause (L') precedes coreferent noun in higher clause (H): When *she* returned *Mary* kissed John.

4. HL' Coreferent noun in higher clause (H) precedes pronoun in lower clause (L'): *Mary* kissed John when *she* returned.

Due to the relatively free positioning of adverbial modifiers in English, the examples do not require the use of passive for constructing the constellations relevant for the Langacker-Ross constraint – in contradistinction to the constructions analyzed in the previous two sections.

The examples 1 and 2 are based on the following DBS graph analysis:

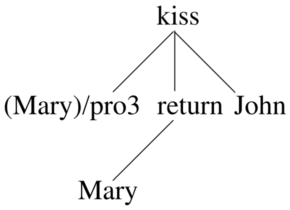
[10] In the variant without the which, transition 8 is an empty traversal.

[11] For a discussion of clausal adverbials apart from the interpretation of pronouns see NLC Sect. 7.5.

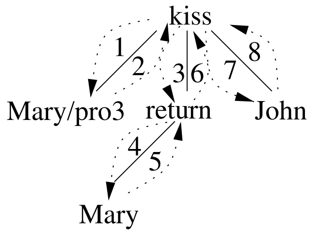
11.5.2 Clausal adverbial modifier: pronoun in higher clause

1. **LH':** When *Mary* returned *she* kissed John.
2. **H'L:** % *She* kissed John when *Mary* returned.

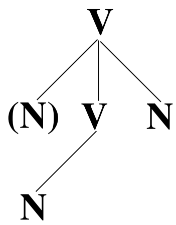
(i) *SRG (semantic relations graph)*



(iii) *NAG (numbered arcs graph)*



(ii) *signature*



(iv) *surface realization*

a.

3	4	5	6	1	2	7	8
When	Mary	returned		she	kissed	John	•
V|V	V/N	N/V	V|V	V\N	N/V	V\N	N\V

b.

1	2	7	8	3	4	5	6
% She	kissed	John		when	Mary	returned	•
V/N	N/V	V\N	N\V	V|V	V/N	N/V	V|V

As expected, the LH' surface (a) has an indexical and a coreferential interpretation. This is expressed by the associated set of proplets:

11.5.3 Content of surface (a) as a set of proplets

noun: Mary	verb: return	noun: (Mary 39)/pro3	verb: kiss	noun: John
cat: snp	cat: #n′ v	cat: s3	cat: #n′ #a′ decl	cat: snp
sem: nm f sg	sem: *when* past	sem: f sg	sem: past	sem: nm m sg
fnc: return	arg: Mary	fnc: kiss	arg: (Mary 39)/pro3 John	fnc: kiss
prn: 39	mdd: (kiss 40)	prn: 40	mdr: (return 39)	prn: 40
	prn: 39		prn: 40	

The pragmatic ambiguity is represented by the (Mary 39)/pro3 value.

The H'L surface (b), in contrast, is limited to an indexical interpretation of the pronoun, expressed by the unambiguous value pro3:

11.5.4 Content of surface (b) as a set of proplets

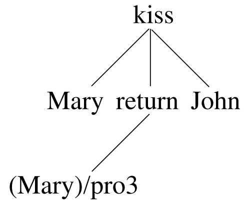
noun: pro3	verb: kiss	noun: John	noun: Mary	verb: return
cat: s3	cat: #n′ #a′ decl	cat: snp	cat: snp	cat: #n′ v
sem: f sg	sem: past	sem: nm m sg	sem: nm f sg	sem: *when* past
fnc: kiss	arg: pro3 John	fnc: kiss	fnc: return	arg: Mary
prn: 40	mdr: (return 39)	prn: 40	prn: 39	mdd: (kiss 40)
	prn: 40			prn: 39

Examples 3 and 4 each allow an indexical and a coreferential interpretation, also as expected:

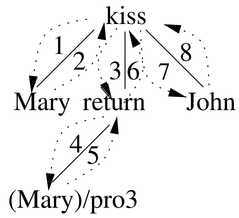
11.5.5 Clausal adverbial modifier: pronoun in lower clause

3. **L'H:** When *she* returned *Mary* kissed John.
4. **HL':** *Mary* kissed John when *she* returned.

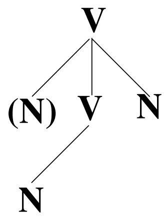
(i) *SRG (semantic relations graph)*



(iii) *NAG (numbered arcs graph)*



(ii) *signature*



(iv) *surface realization*

a.

3	4	5	6	1	2	7	8
When	she	returned		Mary	kissed	John	•
V|V	V/N	N/V	V|V	V/N	N/V	V\N	N\V

b.

1	2	7	8	3	4	5	6
Mary	kissed	John		when	she	returned	•
V/N	N/V	V\N	N\V	V|V	V/N	N/V	V|V

The content of variants (a) and (b) is defined as the following set of proplets:

11.5.6 Content of surfaces (a) and (b) as a set of proplets

noun: (Mary 39)/pro3	verb: return	noun: Mary	verb: kiss	noun: John
cat: s3	cat: #n′ v	cat: snp	cat: #n′ #a′ decl	cat: snp
sem: f sg	sem: past	sem: nm f sg	sem: past	sem: nm m sg
fnc: return	arg: (Mary 39)/pro3	fnc: kiss	arg: Mary	fnc: kiss
mdr:	mdd: (kiss 39)	mdr:	mdr: (return 40)	mdr:
prn: 40	prn: 40	prn: 39	prn: 39	prn: 39

Both variants have both interpretations, as indicated by the (Mary 39)/pro3 values.

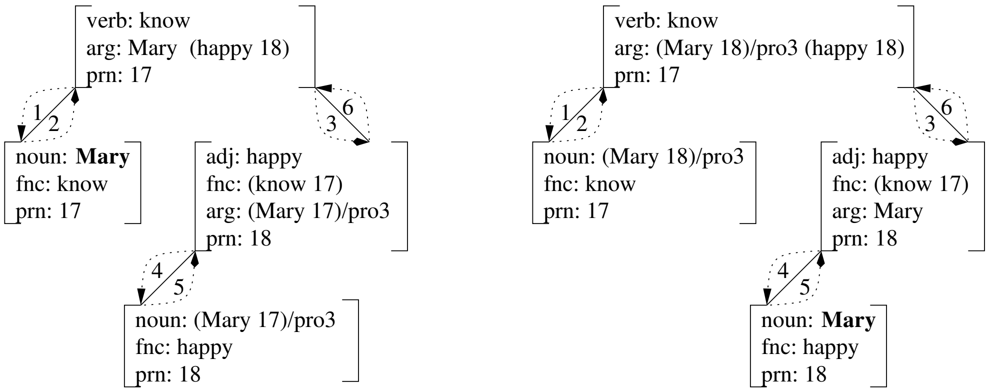
11.6 Structural Explanation of Langacker-Ross Constraint

The Langacker-Ross constraint raises the question of why the subclause constructions LH', L'H, and HL' allow a coreferential interpretation in addition to the indexical interpretation (pragmatic ambiguity), while H'L does not. This question holds equally for extrapropositional functor-argument constructions with a (i) subject clause (11.3.1), (ii) an object clause (11.3.4), (iii) an adnominal modifier clause (11.4.1), and (iv) an adverbial modifier clause (11.5.1).

To explain the assymmetry, let us consider the following proplet NAGs for the L'H (11.3.6) and H'L (11.3.7) constellations of an object clause construction (the explanation transfers in full to the other three constructions):

11.6.1 L'H VS. H'L NAG ALTERNATIVE FOR OBJECT SENTENCE

pronoun in preceding lower clause: **L'H** *pronoun in preceding higher clause:* **H'L**



(iv) surface realization

a:

3	4	5	6	1	2
That	she	was__happy	was__known	to__Mary	.
V|V	V/N	N/V	V|V	V/N	N/V

b:

1	2	3	4	5	6
% She	knew	that	Mary	was_happy	.
V/N	N/V	V|V	V/N	N/V	V|V

The NAGs show the *she* proplet with a double core value, e.g. (Mary 18)/pro3, as if both NAGs offered a choice between an indexical and a coreferential interpretation. In fact, however, the coreferential interpretation is an option only for L'H (passive), but not for H'L (active), as indicated by the %-marker of the surface realization (b).

Ambiguity arises only in the hear mode. Furthermore, the interpretation by the hearer is successful only if it equals the intention of the speaker. If we assume a cognitive agent which is equally competent in the hear and the speak mode, readings which cannot be intended in the speak mode for structural reasons may be excluded from consideration in the hear mode.

In the L'H NAG on the left, the arg value relevant for the coreferential interpretation is Mary, i.e. the intrapropositional subject of the main clause. This value is activated when it is traversed via arcs 1 and 2. Thus when the navigation underlying the speak mode reaches the noun proplet representing she in transition 4, there is an activated referent available for its address value (Mary 17). This explains why a coreferential interpretation is possible in this case.

In the H'L NAG on the right, the corresponding arg value of the top verb is (Mary 18)/pro3, whereby (Mary 18) is the core value option assumed for the sake of the argument. When the proplet representing she is reached in transition 1, the address core value (Mary 18) is an extrapropositional address which is itself in need of a referent. Therefore, the only choice is an indexical pro3 interpretation, pointing at *Suzy*, for example.[12]

Thus, even if a H'L NAG provides a grammatically compatible postcedent for the pronoun, it can not be selected for a coreferential interpretation in the speak mode because the postcedent is not activated in time.[13] This holds for the H'L constellation in all four kinds of extrapropositional functor-arguments.[14]

12 Due to the strictly time-linear derivation order of LA grammar, a delayed surface realization of an extrapropositional address value from a referent to be activated later, here the core value Mary reached in arc 4, is not possible.

13 As an apparent exception, MacWhinney (1999) shows that an H'L construction allows a coreferential interpretation if the preceding main clause is marked as background information by particles:

She$_i$ had *just* returned home when Mary$_i$ saw that the mailbox was open.

As part of the background information, the pronoun she in the mainclause may be assumed to have an antecedent in a preceding sentence.

14 Intrapropositional coreferential pronoun interpretations, for example of reflexives, are another, simpler matter (Helfenbein 2005).

Part III

From Foundations to Applications

12. How DBS Evolved

The starting point in the development of DBS was the linguistic study content taught at university, here Montague grammar.[1] Compared to the dominant schools of nativism, Montague grammar was shown preferable because it offered a formal representation of meaning in the form of truth conditions defined relative to a set-theoretic model structure.

Seemingly the only well-defined formal semantics possible, truth-conditional semantics has since been absorbed into the linguistic main stream. What more could one wish for than a formal grammar of a natural language with a semantic interpretation? The real question, however, is whether such a system allows to reconstruct the mechanism of free natural language communication, verified in the form of a talking robot.

Designed long before the advent of computers, truth-conditional semantics is *sign-based*: logicians use a metalanguage to define set-theoretic models which contain the language signs and an artificial world, and define reference as a direct relation between the signs and this world. A talking robot, in contrast, requires an *agent-based* approach: the real world is treated as given; the robot interacts with it via its external interfaces and internal cognition.

12.1 Ontology

The dichotomy between a sign- and an agent-based approach is a question of ontology in the sense of philosophy.[2] A sign-based approach analyzes expressions of natural language in isolation. An agent-based approach, in contrast, models the human prototype in terms of (i) the hear, (ii) the think, and (iii) the speak mode (Sect. 3.3).

[1] Thanks to Helmut Schnelle, TU Berlin (1968–1970), and to Stanley Peters, UT Austin (1970–1974).

[2] Thanks to Nuel Belnap and Rich Thomason at the Philosophy Department of the University of Pittsburgh (1978–1979), and to Julius Moravscik and Georg Kreisel at the Philosophy Department of Stanford University (1979–1980, 1983–1984) for revealing the secrets of a Tarskian semantics. Later, at Carnegie Mellon University (1986–1989), understanding this fundamental topic was helped further by Dana Scott (FoCL Chaps. 19–21). The 1978–1979 stay in Pittsburgh and the 1979–1980 stay at Stanford were supported by a two-year DFG Research Grant.

In an initial attempt to use truth-conditional semantics for an agent-based approach, SCG (1984) proposed to combine two set-theoretic models inside the agent's head, one representing the language meaning, the other the context of interpretation. The resulting [+sense, +constructive] ontology (FoCL Sect. 20.4) accommodates the First Principle of Pragmatics (PoP-1, FoCL 4.3.3; NLC 2.6.1) and the Principle of Surface Compositionality (1.4.3, FoCL 4.5.1; NLC 1.6.1), at least conceptually.

Seizing the opportunity to use the computers at Stanford University, we attempted to computationally verify the SCG fragment of English with an implementation in Lisp.[3] This seemed possible for the following reasons:

12.1.1 Why programming the SCG fragment seemed possible

1. The SCG fragment is strictly formalized in accordance with Montague grammar, widely considered the highest standard of formal explicitness in truth-conditional semantics.
2. Based on the distinction between the literal meaning_1 of a language sign and the speaker meanings_2 of its use in utterances (PoP-1), SCG analyzes the meaning_1 of a small fragment of English, leaving meaning_2 aside.
3. Transformations and other operations known to increase complexity to exponential or undecidable are excluded from the syntactic-semantic analysis of the language signs by the Principle of Surface Compositionality.

Unfortunately, however, the SCG fragment turned out to be seriously unsuitable for a computational implementation.

The immediate problem was the formalism of categorial grammar[4] (C grammar), which is part and parcel of Montague grammar. Instead of processing input one word after another, C grammar takes a complete sentence as input and starts a bottom-up derivation somewhere in the middle. Thereby it is not obvious whether or not a correct initial composition exists somewhere in the input chain and if so, where; this has the character of problem solving even for experts. Also, C grammar requires a high degree of lexical ambiguity for coding alternative word orders into alternative categories (FoCL Sect. 7.6).

The under-determined derivation order of C grammar causes computational inefficiency, while the absence of a time-linear derivation order makes it incompatible with an agent-based approach. As no time-linear, computationally

[3] Thanks to Stanley Peters and the CSLI Stanford (1984–1986), who made programming the time-linear NEWCAT parser possible.

[4] Designed by Leśniewski (1929) and Ajdukiewicz (1935), the combinatorics of C grammar are coded into lexical categories, using only two canceling rules in a nondeterministic bottom-up derivation order (FoCL Sect. 7.4). This is elegant and simple intuitively, but the derivation order is underspecified by the algorithm and relies heavily on human intelligence (FoCL Sect. 7.5).

efficient alternative was available, we had no choice but to define a new algorithm from scratch (TCS). Called Left-associative grammar (LAG), it was implemented in LISP (NEWCAT 1986).[5] LA grammar is based on the strictly time-linear derivation order and defines the first, and so far the only, complexity hierarchy which is orthogonal to the Chomsky hierarchy. The subclass of C1 LAGs parses the natural languages in linear time (FoCL 12.5.7, 21.5.2).

The failed attempt at programming the Montague grammar defined in SCG corroborated the general insight that a rigorous formalization is a necessary, but not a sufficient, condition for ensuring a reasonable implementation as a well-designed computer program. For any comprehensive linguistic approach, long-term software upscaling will quickly run into unsurmountable difficulty unless there is a declarative specification which integrates the interfaces, the data structure, the database schema, the component structure, the functional flow from input to output, and the motor driving the derivation, based on a reasonable idea of how natural language communication works in principle.

This is confirmed by the painful, long-term upscaling failures of today's sign-based theories in linguistics. Deriving possible substitutions, they (mis)-treat the time-linear structure of language expressions as the "problem of serialization." Without interfaces, their algorithms fail to construct a functional flow from external input to external output. Without an agent-internal memory, there is no place for storing, retrieving, and processing content, such that an essential precondition for a computational reconstruction of the speak, the think, and the hear mode is missing.

12.2 The Functional Flow from Recognition to Action

The functional flow of sign-based C grammar is motivated by hierarchical *constituent structures*, which are also used in phrase structure grammar (PS grammar).[6] Founded on the principle of possible substitutions,[7] context-free PS grammar generates different expressions from a single node, usually called S – without external interfaces and without any external input and output.

Formally, constituent structures are defined in terms of context-free phrase structure trees which fulfill the following conditions:

[5] The stays at Stanford Philosophy (1983–84) and the Stanford CSLI (1984–86), and at CMU computer science (1986–87) and the CMU CMT (1987-88) were supported by a five year Heisenberg Grant (DFG). The 1988–1989 stay at the CMU LCL (Dana Scott and David Evans) was supported by a position as Research Scientist.

[6] As proven by Gaifman in June 1959, bidirectional C grammar and context-free PS grammar are weakly equivalent (Bar Hillel 1964, p. 103). See also Buszkowski (1988) and FoCL Sect. 9.2.

[7] PS grammar uses the principle top down, while C grammar uses it bottom up (FoCL 10.1.6).

12.2.1 Definition of constituent structure[8]

1. Words or constituents which belong together semantically must be dominated directly and exhaustively by a node.
2. The lines of a constituent structure may not cross (non-tangling condition).

According to this definition, one of the following two phrase structure trees is a linguistically correct analysis, while the other is not:

12.2.2 Correct and incorrect constituent structure analysis

claimed to be correct *claimed to be incorrect*

The "correct" tree expresses the intuition that the words knows and John belong closer together semantically than the words Julia and knows.[9]

On the one hand, from a formal point of view both phrase structure trees are equally well-formed. On the other hand, the number of formally possible trees grows exponentially with the length of the sentence.[10] Such a large number of well-formed phrase structure trees for one and the same sentence would be meaningless descriptively if they were all linguistically correct.

Therefore the linguistic concept of constituent structure as defined in 12.2.1 is crucial for any phrase structure analysis, be it in PS grammar or in C grammar: it is the only intuitive principle[11] widely accepted in nativism for excluding most of the formally possible trees.

Yet it has been known at least since 1953 (Bar-Hillel 1964, p. 102) that there are certain natural language constructions, called "discontinuous elements," which violate the definition 12.2.1 of constituent structure. In other words, constituent structure fails to always fit the data. Thus, the second reason why

[8] Ivan Sag,, Stanford 1985.

[9] To someone not steeped in nativist linguistics, this intuition may be difficult to follow. It is related to the substitution and movement tests of American structuralism (FoCL Chap. 8).

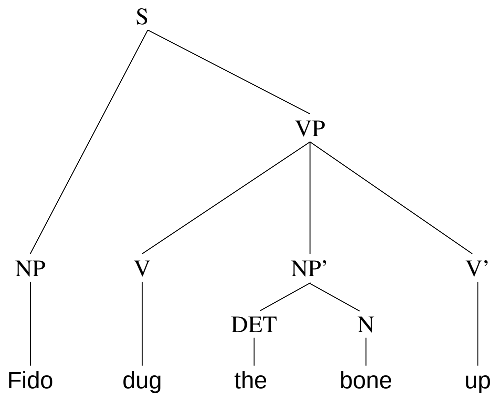
[10] If loops like A → ...A... are permitted in the rewrite rules (which they usually are), the number of different possible trees over a finite sentence is infinite!

[11] Historically, the definition of constituent structure is fairly recent; it goes back to the *immediate constituent* (IC) analysis of Bloomfield (1933).

the Montague grammar defined in SCG had to be abandoned, apart from the computationally inadequate derivation order of C grammar, is the empirical inadequacy of constituent structure.

Known as the Constitutent Structure Paradox (FoCL Sect. 8.5), the problem may be illustrated with discontinuous dug__up in two vain attempts to analyze Fido dug the bone up as a well-formed constituent structure:

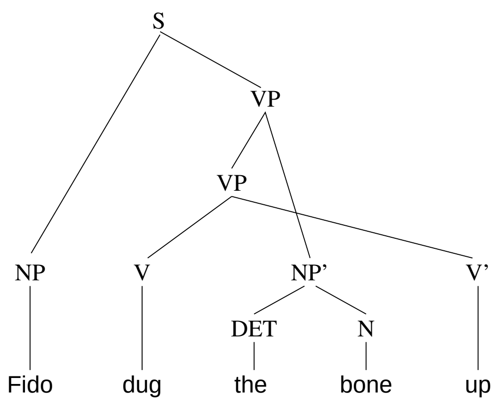
12.2.3 First attempt: violating condition 1



The lines do not cross, satisfying the second condition of 12.2.1. The analysis violates the first condition, however, because the semantically related expressions dug__ up (discontinuous element), or rather the nodes V (verb) and V' dominating them, are not dominated *exhaustively* by a node. Instead, the VP node directly dominating V and V' also dominates the NP' the bone.

There is another formal possibility, which fails as well:

12.2.4 Second attempt: violating condition 2



Here semantically related dug and up are dominated directly and exhaustively by a node, thus satisfying the first condition of definition 12.2.1. The analysis violates the second condition, however, because the lines in the tree cross.

Discontinuous elements are a problem for constituent structures because the latter are defined in terms of context-free PS grammar (FoCL Sect. 8.1). The only relations generated by this class is *pairwise inverse*, e.g. abc...cba, which is of n^3 complexity (polynomial). Relations which are not pairwise inverse, such as abc...abc (not inverse, FoCL 11.5.6) and aaabbbccc (not pairwise, FoCL 10.2.3), fall into the next higher class, called context-sensitive. This class is of exponential complexity, which is computationally intractable.

The distinction between context-free and context-sensitive relations is an artefact of the substitution-based PS grammar rule formats (FoCL 8.1.2) and has no empirical foundation in natural language. Even though the $aba'b'$ structure of discontinuous elements is not context-free, they occur frequently, not just in a few natural languages, but in many, if not all. An example of an ubiquitous discontinuous construction is the perfect tense of the transitive verb in declarative main clauses in German, as in Peter *hat* das Buch *gelesen*.

Because context-free PS grammar can handle only binary inverse relations, about half of today's linguists interested in this matter believe that the computational complexity of natural language is exponential.[12] The other half declares the natural languages context-free because that is the largest class in the PS grammar hierarchy which is still computationally tractable.

In LA grammar, the paradox does not arise. Instead of building constituent structure with direct and exhaustive dominance (12.2.1), LA grammar uses the strictly time-linear[13] derivation order in combination with the classic relations of functor-argument and coordination. LA grammar parses context-sensitive $a^kb^kb^k$ in linear and WW (FoCL 11.5.5) in n^2 polynomial time; "discontinuous elements" have no effect on these low complexity degrees (12.3.3).

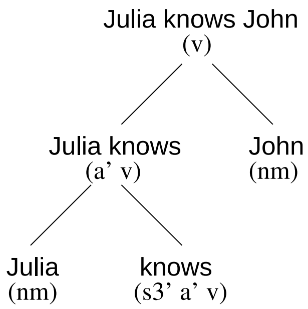
12.3 Computing Possible Continuations

The following reanalysis of example 12.2.2 replaces the underspecified[14] derivation order of substitution-based grammars with the strictly time-linear derivation order of continuation-based LA grammar. Consider the following derivation structure, to be read bottom-up:

[12] The complexity of transformational grammar is even higher than exponential (FoCL 8.2.3), namely undecidable (Peters and Ritchie 1973).

[13] This is in concord with de Saussure's (1913/1972) Second Law. Time-linear means linear like time and in the direction of time – in contradistinction to linear time, which is the lowest basic degree of

12.3.1 Conceptual NEWCAT analysis of Julia knows John



Based on valency canceling (Tesnière 1959), the time-linear analysis combines a *sentence start* and a *next word* into a new sentence start. For example, the sentence start [Julia (nm)] and the next word [knows (s3′ a′ v)] combine into the new sentence start [Julia knows (a′ v)]. The category segment nm (for name) serves as a filler which cancels the valency position s3′ in the category of knows.

In NEWCAT, the computation of possible continuations doubles as the automatic grammatical analysis: the software operations are displayed as a suitably formatted *trace* in the sense of computer science.[15] The result is absolute 'type transparency' (FoCL Sect. 9.3) in the sense of Berwick and Weinberg (1984).

12.3.2 Automatic NEWCAT parse of example 12.3.1

```
NEWCAT>  Julia knows John \.

    *START
    1
       (SNP) JULIA
       (N A V) KNOWS
    *NOM+FVERB
    2
       (A V) JULIA KNOWS
       (SNP) JOHN
    *FVERB+MAIN
    3
       (V) JULIA KNOWS JOHN
       (V DECL) .
    4
    *CMPLT
       (DECL) JULIA KNOWS JOHN .
```

computational time complexity (FoCL Sect. 8.2).

[14] Nativism treats derivation order as a "performance" phenomenon, irrelevant for "competence." The partial order of possible substitutions is reflected in the many possible derivation orders for parsing phrase structure trees, like left-corner, right-corner, island, etc. Possible substitutions express the nativist view of grammar as a *generation* mechanism, like describing the growth of a plant from a single seed. Unfortunately, this is the wrong model for a language-based transfer of content between agents.

[15] The NEWCAT (1986) parser covers 221 grammatical constructions of German and 114 of English.

The comparatively vague intuitions about what "belongs semantically together" (which underlie the definition of constituent structure 12.2.1) are replaced by the classic semantic relations of functor-argument and coordination, and coded in categories defined as lists of one or more category segments. In this way, the constituent structure paradox is avoided, as shown by the following NEWCAT parse:

12.3.3 NEWCAT PARSING OF Fido dug the bone up

```
NEWCAT>  Fido dug the bone up \.

    *START
    1
       (SNP) FIDO
       (N A UP V) DUG
    *NOM+FVERB
    2
       (A UP V) FIDO DUG
       (SN SNP) THE
    *FVERB+MAIN
    3
       (SN UP V) FIDO DUG THE
       (SN) BONE
    *DET+NOUN
    4
       (UP V) FIDO DUG THE BONE
       (UP) UP
    *FVERB+MAIN
    5
       (V) FIDO DUG THE BONE UP
       (V DECL) .
    *CMPLT
    6
       (DECL) FIDO DUG THE BONE UP .
```

The discontinuous element up is treated as a filler for the valency position up′ in the lexical category (n′ a′ up′ v)[16] of dug. The phrasal noun the bone is added in two time-linear steps: the article the has the category (sn′ snp) such that the category segment snp cancels the valency position a′ in the category (a′ up′ v) of Fido dug, while the category segment sn′ is added in the result category (sn′ up′ v) of Fido dug the. In this way the obligatory addition of a noun after adding the determiner the in English is ensured.

The Lisp source code was re-implemented by several NEWCAT readers in South America, Switzerland, and other locations.[17] Later, an algebraic definition was distilled from the Lisp code.[18]

[16] The use of ′ to distinguish a valency position, e.g. a′, from a valency filler, e.g. a, was introduced later. The proper treatment of grammatical number in definite noun phrases (NLC) had not yet been found.

12.4 Deriving Content

The design of the NEWCAT parser solved three problems. First, by replacing the underspecified derivation orders of C and PS grammar with a strictly time-linear derivation order, we arrived at the new algorithm of LA grammar, which parses the natural languages in linear time (FoCL Sects. 12.5 and 21.5). Second, the time-linear building up and canceling of valency positions provides a semantically motivated analysis of natural language which eliminates the Constituent Structure Paradox (Sect. 12.2). Third, the failure of C and PS grammar to be I/O equivalent with the human prototype is repaired by using the algorithm of NEWCAT for the hear mode, which takes a sequence of modality-dependent unanalyzed external word form surfaces as input.

What was missing, however, was a derivation of *content* in the hear mode, to be processed in the think mode, and used as input to the speak mode. As in C grammar, NEWCAT derivations of different declarative sentences all end in the same category, namely (v),[19] i.e. a verb without any unfilled valency positions.[20] The question was how to provide the NEWCAT parsing of natural language with a semantic representation of (i) language content (meaning$_1$) which is suitable also for building (ii) the context of interpretation, as a structural precondition for (iii) a simple pattern matching between the language content and the context content in accordance with PoP-1 (1.4.1).

The most main stream solution at the time would have been a truth-conditional approach. However, while attempting to program the SCG fragment at CSLI Stanford in 1984, it had already become clear that two set-theoretical models (one representing the meaning$_1$, the other the context of use) are not suitable[21] for a cognitive modeling of reference as a computationally viable pattern matching (FoCL 22.2.1).

The perhaps second most main stream solution at the time was in the form of phrase structure trees, interpreted as semantic hierarchies. This was attempted in CoL (1989). Proceeding along the then popular separation of syntax and

[17] These efforts became known because one little function had been omitted accidentally in the NEWCAT source code publication, causing several of the re-programmers to write and ask for it.

[18] Thanks to Stuart Shieber and Dana Scott, who at different times and places helped in formulating the algebraic definition for LA grammar, published in CoL (1989) and TCS (1992).

[19] Disregarding the addition of the period (12.3.2, 12.3.3).

[20] This is similar to Montague grammar, in which the derivation of different declarative sentences all end in the category t (for true) if successful.

[21] Thus, even if a C grammar with lambda reduction were to derive set-theoretical models from language expressions in a time-linear derivation order (as proposed by Kempson et al. 2001), these models would not be practical for a [+sense, +constructive] (FoCL 20.4.2) reconstruction of reference in natural language communication because computational pattern matching remains unsolved.

semantics ("autonomy of syntax"), the basic idea was to use the NEWCAT parser as a syntactic algorithm and define a semantic interpretation for it.[22]

For complementing NEWCAT derivations with a parallel construction of semantic hierarchies we used the FrameKit+ software (Carbonell and Joseph 1986). The resulting software provided the automatic NEWCAT analyses with homomorphic semantic hierarchies in real time. For CoL, the NEWCAT fragment was extended to 421 grammatical constructions of English.

Consider the semantic interpretation of the NEWCAT analysis 12.3.3 in CoL (p. 44):

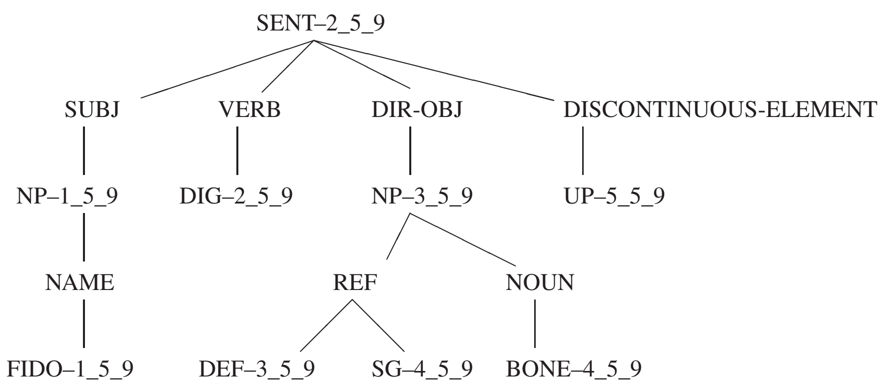
12.4.1 Semantic interpretation as a frame structure in CoL

```
Hierarchical Analysis:

 (SENT-2_5_9
  (SUBJ ((NP-1_5_9 (NAME (FIDO-1_5_9)))))
  (VERB (DIG-2_5_9))
  (DIR-OBJ ((NP-3_5_9 (REF (DEF-3_5_9 SG-4_5_9))
                                 (NOUN ((BONE-4_5_9))))))
  (DISCONTINUOUS-ELEMENT ((UP-5_5_9))))
```

Using a suitable tree-printer, this structured list may be automatically displayed as the following semantic hierarchy (CoL, p. 45):[23]

12.4.2 Displaying the frame structure 12.4.1 as a tree



The time-linear syntactic analysis 12.3.3 and its semantic interpretation 12.4.1/12.4.2 are derived simultaneously in real time.

[22] Thanks to Jaime Carbonell and the CMT/LTI at Carnegie Mellon University (1986–1989), who made programming the semantically interpreted CoL parser possible. Thanks also to Brian MacWhinney, who supported a fruitful three-month stay at the CMU Psychology Department in the fall of 1989.

[23] The tree printer available at the time connects nodes with the corners ⌜ and ⌝ in typewriter font, as shown in TCS Fig. 2.1. For better readability, the corners have been replaced here by the diagonal lines familiar from PS trees.

Superficially, 12.4.2 may seem to resemble a constituent structure, but neither its intuitive assumptions nor its formal definition 12.2.1 are satisfied. In particular, the assumption that dug is semantically closer to the bone than to the dog is not expressed.[24] Nota bene: all constituent structures are semantic hierarchies, but not all semantic hierarchies are constituent structures.

The automatic interpretation of a substantial fragment of a time-linear NEWCAT syntax with a homomorphic semantics of a traditional kind provided the CoL approach with proof of concept. The next question was *what to do with it?*. Stolidly expanding the data coverage of CoL for no other purpose than storage in some "tree bank" was not appealing. A better use of real time CoL parsing seemed the interpretion and production of living language, on the fly, as in the spontaneous dialog of free human-machine communication.

We began with another attempt at treating reference as a pattern matching, this time between two frame structures, one representing the meaning$_1$, the other the context of use. It turned out, however, that conventional frames (Carpenter 1992) – like the set-theoretical models used in SCG (Montague 1974) – are not suitable for a software-mechanical pattern matching (FoCL 22.2.2).

12.4.3 Basic problems for matching with conventional frames

1. A content may be coded as different, but equivalent, frame structures; such variations[25] obstruct pattern matching when it should be possible.
2. The recursive embedding of frames (12.4.1), used for establishing semantic relations, complicates a computationally viable matching.

Thus the CoL project had to be abandoned. In a new effort, problem (1) was solved by representing a content as an *order-free*[26] set of proplets. Instead of connecting the elements of a proposition by their position in a formula, behind a quantifier, in a tree, or in a frame, the proplets of a proposition are order-free and connected by a common prn (proposition) number.

Problem (2) was solved by the definition of proplets as *non-recursive* feature structures. Instead of representing functor-argument structure by embedding

[24] Context-free PS grammar formalizes this asymmetry (12.2.2) in the rules S $\rightarrow$ NP VP and VP $\rightarrow$ V NP. DBS, in contrast, follows the logical tradition by treating the subject and a possible object as equal arguments, as in f(a, b) .

[25] Consider $(p \vee q)$ and $(q \vee p)$ as propositional contents. Because they are defined to be semantically equivalent (axiom of symmetry), they should match, but their structural difference requires extra work. In DBS, this problem is solved by recoding formulas of propositional calculus as order-free sets of proplets which specify inter-proplet relations by means of addresses (Hausser 2003).

[26] The assumption of an order-free content seems to agree with the SemR level of Meaning-Text theory (MT) as proposed by Zholkovskij and Mel'chuk (1965). An order-free level is also used in some dependency grammars, e.g. Hajičová (2000).

the feature structure of the argument(s) into the feature structure of the functor,[27] as in nativism, the classic relations of functor-argument and coordination are coded alike as proplet-internal addresses and implemented as pointers.

The matching between a pattern proplet and a content proplet is straightforward and efficient because of the fixed order of attributes and the use of flat features. The matching of values is based on (i) types matching tokens (NLC Sect. 4.1), and on (ii) variables matching constants (Sects. 3.2, 4.3).

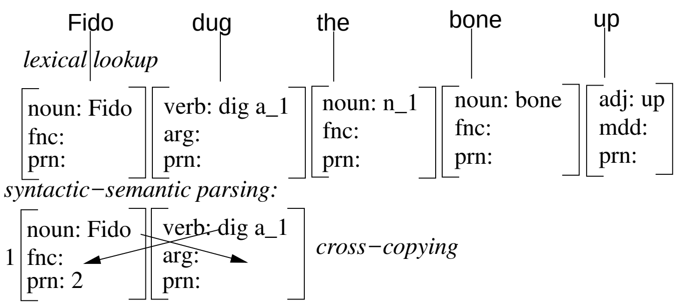
Pattern matching is the basic mechanism of DBS. It is used (i) in reference as a matching between a language meaning$_1$ and a context of interpretation (NLC 3.2.4), (ii) in the recognition and action procedures of peripheral cognition (Sect. 8.5; NLC 4.5.2), and (iii) in applying the operations of DBS.Hear, DBS.Nav, DBS.Inf, and DBS.Speak to input content.

12.5 Graphical Representations in DBS

For theoretical and practical reasons, we developed DBS and the SLIM theory on which it is based (FoCL Sect. 4.2; NLC Sect. 2.6) as a *declarative specification* in the sense of computer science (NLC 1.2.1). A declarative specification may be defined in terms of formal rules, for example, a logical production system or an LA grammar. A sometimes more intuitive approach, however, is graphical. This holds especially for the characterization of the component structure and the functional flow in a complex system. It holds also for characterizing the derivations in the hear, the think, and the speak modes of DBS.

Compared to representing a hear mode derivation as a trace of the NEWCAT parser (12.3.2, 12.3.3), the NLC format based on the data structure of proplets is more comprehensive. Consider the graphical format of NLC in the hear mode derivation of our discontinuous element example:

12.5.1 NLC HEAR MODE DERIVATION OF Fido dug the bone up



[27] Recursive embedding cannot even be properly extended to coordination – which is the other semantic relation of structure in Aristotelian semantics besides functor-argument.

2	noun: Fido fnc: dig a_1 prn: 2	verb: dig a_1 arg: Fido prn: 2	~~noun:~~ n_1 fnc: prn:		*cross−copying*
3	noun: Fido fnc: dig a_1 prn: 2	verb: dig a_1 arg: Fido n_1 prn: 2	noun: n_1 fnc: dig a_1 prn: 2	~~noun:~~ bone fnc: prn:	*absorption with simultaneous substitution*
4	noun: Fido fnc: dig a_1 prn: 2	verb: dig a_1 arg: Fido bone prn: 2	noun: bone fnc: dig a_1 prn: 2		
5	noun: Fido fnc: dig a_1 prn: 2	verb: dig a_1 arg: Fido bone prn: 2	noun: bone fnc: dig a_1 prn: 2	~~adj:~~ up mdd: prn:	*absorption with simultaneous substitution*

result of syntactic−semantic parsing:

noun: Fido fnc: dig up prn: 2	verb: dig up arg: Fido bone prn: 2	noun: bone fnc: dig up prn: 2

The derivation is read top-down, like 12.3.3, – in contradistinction to 12.3.1, which is read bottom-up like the trees of C grammar (FoCL 7.4.5, 7.5.3).

The result of the derivation 12.5.2 may be shown in more detail as the following set of proplets:

12.5.2 Representing a content as a set of proplets

sur: noun: Fido cat: snp sem: nm fnc: dig up mdr: nc: pc: prn: 2	sur: verb: dig up cat: #n′ #a′ decl sem: past arg: Fido bone mdr: nc: pc: prn: 2	sur: noun: bone cat: snp sem: def sg fnc: dig up mdr: nc: pc: prn: 2

In contradistinction to C grammar and NEWCAT, valency canceling is not treated as the deletion of valency positions, but as #-marking them, as shown by the cat feature of the verb. This is because the information must be preserved for word form production in the speak mode (NLC Chaps. 12, 14).

The categorial operations based on processing the proplets' cat values continue the earlier NEWCAT technique to build up and cancel valency position (12.3.2, 12.3.3). However, they are now complemented by the *cross-copying* of values, shown in 12.5.1 by diagonal arrows. Cross-copying values between proplets builds up content as a set of items which are order-free but connected semantically by means of addresses, implemented as pointers. The result of a hear mode derivation is thus ideally suitable for storage and retrieval (activation) in a data base, here a content-addressable word bank (4.1.1).

Complementary to the design of a graphical representation for the hear mode, there remained the task of developing one for the speak mode. In a first attempt (NLC 2006), we instinctively (but erroneously) tried to define LA think and LA speak as separate but interacting, alternating mechanisms: a LA think navigation step from one proplet to a next, along a semantic relation between them, triggered an LA speak application for realizing as many surfaces as supported by the proplets traversed so far. Completion of an LA speak application step in turn triggered a switch back to LA think to continue the navigation.

As an illustration, consider the following characterization of the discontinuous construction familiar from 12.2.3, 12.2.4, 12.3.3, and 12.5.1.

12.5.3 SCHEMATIC NLC PRODUCTION OF Fido dug the bone up.

activated sequence				*realization*
i				
	V			
i.1		n		n
	V	N		
i.2	fv	n		n fv
	V	N		
i.3	fv	n	d	n fv d
	V	N	N	
i.4	fv	n	d nn	n fv d nn
	V	N	N	
i.5	fv de	n	d nn	n fv d nn de
	V	N	N	
i.6	fv de p	n	d nn	n fv d nn de p
	V	N	N	

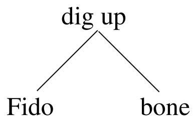
The upper case letters V and N represent verb and noun proplets, respectively. The lower case letters represent abstract word form surfaces: n for name, fv for finite verb, d for determiner, nn for noun, de for discontinuous element, and p for period. The navigation order VNN produces surfaces in a different order, namely n fv d nn de p, representing Fido dug the bone up . (cf. i.6).

What is missing, however, is a characterization of the semantic relations of structure holding between proplets. Also, separate but alternating LA think and LA speak grammars turned out to be prohibitively complicated to program. Therefore the idea of separate LA think and LA speak grammars had to go. As a simpler solution, lexicalization rules (NLC 12.4.3, 12.5.2, 12.6.1) are now embedded into the sur slot of the proplets traversed by DBS.Nav. DBS grammars with language-dependent lexicalization rules are called DBS.Speak.

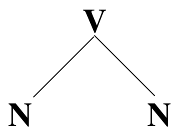
To graphically characterize the semantic relations traversed by DBS.Nav, we adapted an idea of Frege (1878) which has been lauded in the literature but, to the best of our knowledge, has not been taken up to characterize the semantic relations of structure in natural language until now. It assigns unique semantic interpretations to different kinds of lines in a graph.[28] As an example, consider the following DBS graph analysis underlying the speak mode for producing Fido dug the bone up. (TExer Sect. 4.3):

12.5.4 DBS GRAPH ANALYSIS OF A DISCONTINUOUS STRUCTURE

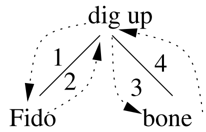
(i) *SRG (semantic relations graph)*



(ii) *signature*



(iii) *NAG (numbered arcs graph)*



(iv) *surface realization*

1	2	3	4
Fido	dug	the_bone	up_.
V/N	N/V	V\N	N\V

The (i) SRG and the (ii) signature show the semantic relations of structure using the line (edge) "/" for the subject/predicate and the line "\" for the object\predicate relation. The arrow numbering in the (iii) NAG is standard pre-order as defined in graph theory. Reused in the (iv) surface realization, the numbering shows that the surface Fido is realized from the goal node of transition 1, dug from 2, the_bone from 3, and up_. from 4.

Not unlike 12.4.2, the graphs (i-iii) in 12.5.4 have a superficial resemblance with a constituent structure. This may be welcome insofar as graphical representations have long been an essential part of linguistic intuitions. It should be clear, however, that the motivation behind the two kinds of graphs is different.

First, a constituent structure tree shows dominance and precedence in a language sign, while a DBS graph shows semantic relations of structure in a content. Second, constituent structure is motivated by the movement and substitution tests of American Structuralism (Bloomfield 1933; Harris 1951), while the graphs of DBS are motivated by the classic functor-argument and coordination relations of natural language, at the elementary, phrasal, and clausal level of grammatical complexity (Sect. 9.6).

[28] A different method with a similar purpose is *labeling* the edges (7.1.1, 7.1.2) in modern Dependency grammar (Liu 2009b).

12.6 Different Applications for the Abstract DBS Agent

As an abstract agent, the software machine of DBS may be applied to behavior modeling in general. Such applications (i) take a time-linear stream of raw data as input, (ii) process them into content which (iii) is stored, (iv) selectively activated, and (v) inferenced upon in a content-addressable memory, and (iv) turned into an output in the form of action vis á vis the current state of the agent's environment. Different applications will use the same algorithm, data structure, and database schema, but may differ in terms of their interfaces, attributes, values, semantic relations, and grammatical complexity levels. They may also be loaded with different agendas for preferred outcomes such as survival of the group, successful reproduction, fame and fortune, or peace and justice.

The notion of an abstract agent in interaction with a constantly changing environment may be applied, for example, to the life cycle of different kinds of fauna[29] and flora, the behavior patterns in social communities, or the evolution of ecosystems. As long as recursive ambiguity (FoCL Sects. 11.2–11.5) is avoided in the input interpretation, the computational complexity of the abstract agent's recognition will be linear (TCS).

While the interfaces of recognition and action, and the functional flow from input to output are concretely given in a talking robot, this may be a challenge in some of the other applications. Consider, for example, applying the notion of an abstract agent to a colony of ants: it requires reconstructing the external interfaces and the functional flow for the whole community, building on the interfaces of the individuals and their communication.[30]

The input and output items of any such DBS application must be concrete (surface compositional) and the attributes, relations, and levels must be limited to those which may be shown to be empirically grounded. In the application of DBS to natural language communication, these methodological desiderata have led us to the following properties:

12.6.1 Number of content elements used by a talking robot

1. The number of available core values is determined by the robot's needs to survive in an environment. It is limited by the agent's external interfaces for recognition and action, and the core value types provided by memory (8.5.1, 8.5.3).
2. The number of the core attributes for building lexical entries is three (Sect. 3.5), represented by the proplet shells for N (noun), V (verb), and A (adj). This continues the traditional distinction in Western logic between

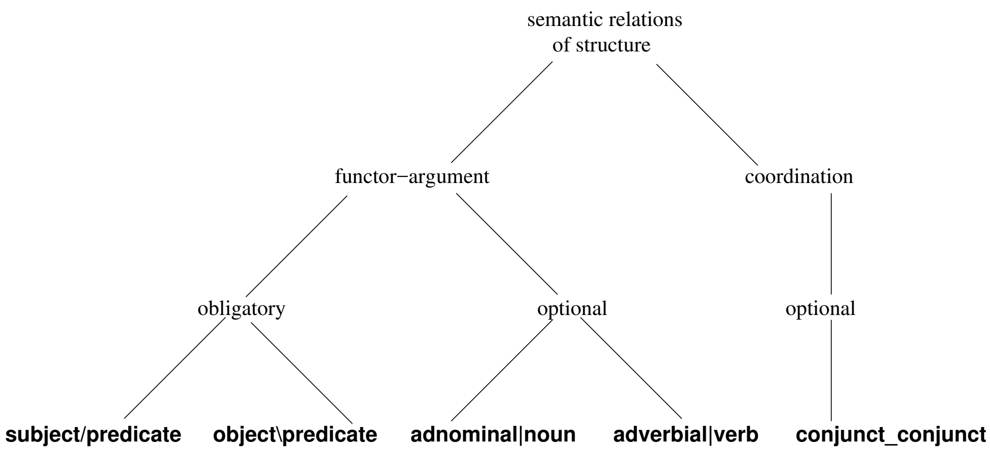
arguments (N), functors (V), and modifiers (A), and in Western philosophy between objects (N), relations (V), and properties (A).

3. The number of semantic relations of structure is five, namely (i) subject/predicate, (ii) object\predicate, (iii) adnominal modifier|modified, and (iv) adverbial modifier|modified for functor-argument, and (v) conjunct−conjunct for coordination. This continues the Aristotelian approach exemplified by the traditions of propositional and predicate calculus (Sect. 13.1).
4. The number of grammatical complexity levels is three, namely (i) phrasal, (ii) elementary, and (iii) clausal (9.6.2). Levels (i) and (ii) constitute the intrapropositional relations, while level (iii) comprises the extrapropositional relations.

The relation between core values (1) and core attributes (2) is orthogonal: a core value may be embedded into different proplet shells and a proplet shell may take different core values (Sect. 6.6). The kind of semantic relation (3) between proplet kinds (2) is partially restricted: while no pairs of proplet kinds are generally excluded from being connected by a relation, the specific relations permitted depend on the language (Chap. 7).

The classic semantic relations of functor-argument and coordination occur in all natural languages (i.e. they are universal) and build language as well as nonlanguage (context) content. Graphically, they may be shown as follows:[31]

12.6.2 Semantic relations of structure as a hierarchy



[29] Reconstructing certain abilities of living beings has long been a topic of cognitive robotics and nouvelle AI.

[30] It remains to be seen whether the nonwritable memory of a fixed behavior agent (Sect. 6.1) will suffice for this application.

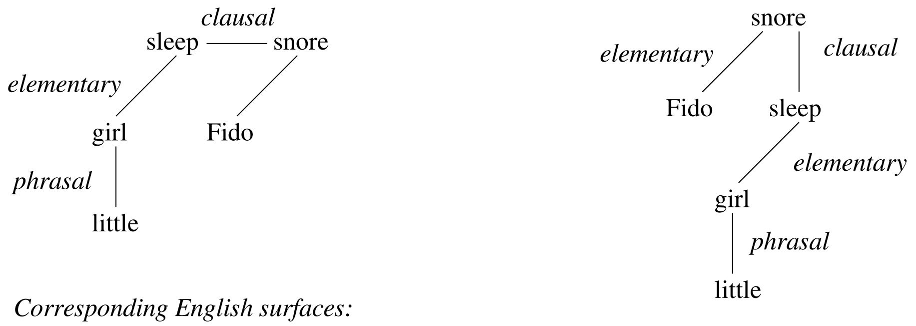
[31] This graph complements 4.6.1 in NLC.

Functor-argument may be obligatory or optional. The obligatory functor-arguments are subject/predicate and object\predicate; they are lexically controlled by the valency positions of the verb. The optional functor-arguments are adnominal|noun and adverbial|verb. Coordination is also optional.

The combination of the five semantic relations of structure (12.6.1, 3) and the three levels of grammatical complexity (12.6.1, 4) allows to construct contents of arbitrary size without recursion[32]: (i) concatenating a main clause with a main clause provides for an unlimited horizontal extension of content and (ii) concatenating a higher clause with a subclause provides an unlimited (at least in theory) vertical extension. These two dimensions of clausal content extension are illustrated by the following DBS graphs (cf. 9.4.1 for the NAGs):

12.6.3 The two dimensions of clausal content extension

horizontal extension (conjunct−conjunct) vertical extension (modifier|modified)



Corresponding English surfaces:
The little girl slept. Fido snored. When the little girl slept, Fido snored.

The horizontal content extension on the left is an extrapropositional coordination, while the vertical content extension on the right is an extrapropositional functor-argument, here an adverbial sentential modifier construction.

[32] There are two related reasons why DBS derivations are not recursive in any formal sense. First, as a data structure, proplets are defined as nonrecursive feature structures. Second, the semantic relations of structure between proplets are realized by address and not by substituting items in a self-similar way, as in x = x+1, NP → NP S, or N → ADJ N.

Operation patterns matching input data allow DBS to derive and process contents as autonomous (unsupervised) self-organizing networks. This is different from the four networks analyzed by Hammer et al. (2004), which "can be extended in a straightforward manner to recursive data structures" (cited from ResearchGate website, p. 65).

13. Semantics

In the logical tradition, a formal language (or a formalized "fragment" of a natural language) is interpreted semantically by combining (i) a set of categorized words (lexicon) and (ii) rules of composition (syntax) with (iii) rules of semantic interpretation. The purpose is (a) to build formal contents and (b) to define inferences on them.

There are several different ways to build a semantically interpreted language (FoCL Chap. 19, Sect. 20.5). They all have in common, however, that two kinds of meaningful elements are distinguished, namely (i) the *logical* (necessary, absolute) and (ii) the *contingent* (possible, accidental).

Examples of logical elements are the connectives, quantifiers, variables, and the bracketing syntax in predicate calculus; they are the basis of inferencing and theorem proving in symbolic logic. Other examples are the attributes and the variables of proplets in DBS; they are used for building patterns (3.2.3, 3.2.6) which provide the structural framework for processing content.

Examples of contingent elements are sign-based constants like man′ or sleep′,[1] defined as truth conditions relative to set-theoretical models in Montague grammar (i.e. a form of predicate calculus in symbolic logic). Other examples are agent-based core values, grounded in the cognitive agent's recognition and action procedures of DBS (Sect. 8.5; NLC 2.3.1).

13.1 The Aristotelian Approach

Symbolic logic and DBS share not only the distinction between logical and contingent elements, but also the property of being Aristotelian. At the core of the Aristotelian approach is the use of only two semantic relations of structure, namely classic (i) *coordination* and (ii) *functor-argument*, at the intrapropositional (elementary and phrasal) and the extrapropositional (clausal) level of grammatical complexity (Sect. 9.6).

[1] The distinction between an expression, e.g. *man*, and its translation into intensional logic, e.g. man′, (Montague 1974, p. 266), is expressed in DBS by a proplet's separate sur and core attributes. This frees the apostrophe for the distinction between a valency position, e.g. a′, and its filler, e.g. a.

In symbolic logic, this fundamental insight is realized by (i) propositional calculus for extrapropositional coordination and (ii) predicate calculus for intrapropositional functor-argument. These may be extended into (iii) extrapropositional functor-argument[2] and (iv) intrapropositional coordination.[3]

Symbolic logic represents extrapropositional coordination with expressions like $((p \wedge q) \vee r)$. They are composed by the syntactic rules of propositional calculus, which syncategorematically[4] introduce the binary operators $\wedge$, $\vee$, and $\rightarrow$ as well as the unary operator $\neg$.

DBS uses function words instead of the logical operators. Analyzed lexically by automatic word form recognition, function words carry their semantic contribution as values inside their proplet representation. An example is the extrapropositional DBS coordination corresponding to f(a)∧f'(a')∧f"(a"), shown concretely as the content 3.2.4 and abstractly as the pattern 3.2.5.

Symbolic logic represents intrapropositional functor-arguments as, e.g., f(a,b) where f is a functor and a and b are arguments. The functor is like a two-place verb, while the arguments are like nouns serving as subject and object.[5]

DBS establishes functor-argument relations by cross-copying the core value of one proplet into the continuation slot of another (3.3.1). As feature structures, proplets are more differentiated and intuitive than a functor like f or the individual constants a_1, a_2, a_3, ... of symbolic logic, without any loss of generality.[6] The intrapropositional DBS functor-argument corresponding to f(a,b) is illustrated concretely as the content 3.2.1 and abstractly as the pattern 3.2.2.

---

[2] However, the donkey sentence 11.4.4 shows that extending predicate calculus extrapropositionally to subclauses is not always possible. This defect has been recognized as serious enough to spawn a massive body of literature trying to repair it.

[3] Extending propositional calculus from extrapropositional to intrapropositional coordination also creates a problem for symbolic logic. It arises with examples like All the students gathered; John and Mary are a happy couple; Suzy mixed the flower, the sugar, and the eggs; etc., which intuitively suggest a collective rather than a distributive reading (see Zweig 2008 for an overview; see also Hausser 1974; Kempson et al. 1981).

Structurally, the mix problem can not occur in extrapropositional coordination, just as the donkey problem can not occur in intrapropositional functor-argument. The mix problem arises in the interaction between an intrapropositional coordination in plural arguments and certain verbs, and may have a tolerable solution within symbolic logic by proposing new quantifiers and/or connectives. The donkey problem, in contrast, is caused by failing quantifier scope and has no clean solution within the hierarchical bracketing structure of predicate calculus. Neither problem arises in DBS because content is coded by means of order-free proplet sets connected by address.

[4] The syncategorematic introduction of an operator avoids its lexical definition. An example is the introduction of $\wedge$ by the rule "If p and q are sentences, then p∧q is a sentence" (FoCL 19.3.2). At the language level, the syncategorematic method violates surface compositionality.

[5] Using typed lambda calculus, Montague (1974) formalized functors and arguments set-theoretically as semantic types with corresponding syntactic categories. Lambda reduction in a typed lambda calculus may be viewed as a souped-up version of the categorial canceling rules (FoCL 7.4.3).

[6] If needed, a feature may be represented abstractly as a pattern in which the attribute and the values are represented by replacement variables (NLC 4.2.3, 4.2.4).

13.2 Reinterpreting Montague Grammar as Hear Mode

The traits common to symbolic logic and DBS may be increased further by taking the liberty to reinterpret the semantic interpretation of a sign-based approach as the hear mode of an agent-based approach.[7] In this way, a sign-based approach may be used to cover one of the three steps of the natural language communication cycle.

This holds especially for Montague (1974), who shows in PTQ how to semantically interpret the surfaces of a formal fragment of English by translating them into formulas of predicate calculus (typed lambda calculus). The translation is quasi-mechanical[8] in that the reader may go mentally through the rule applications and lambda reductions as if going through the steps of a proof.

From the viewpoint of such an agent-based reinterpretation as a hear mode grammar, DBS may be seen as completing Montague grammar in two ways. One is the extension of Montague grammar to the full cycle of natural language communication by adding the think and the speak mode. The other is replacing Montague's quasi-mechanical interpretation method with the computational automation provided by efficiently running software.

These two completions of Montague grammar required agent-based solutions which differ from those of symbolic logic. For example, predicate calculus uses the logical elements to construct a syntactic exoskeleton, while DBS integrates the logical and the contingent elements into nonrecursive feature structures with ordered attributes (proplets) which code inter-proplet relations solely by address, and are therefore order-free (3.2.8).

Another difference is the absence of quantifiers in DBS. To see why let us consider the following example from Montague (1974, PTQ p. 268):

13.2.1 PREDICATE CALCULUS ANALYSIS OF A woman loves every man.

reading 1: $\exists x[\text{woman}'(x) \wedge \forall y[\text{man}'(y) \rightarrow \text{love}'(x, y)]]$

reading 2: $\forall y[\text{man}'(y) \rightarrow \exists x[\text{woman}'(x) \wedge \text{love}'(x, y)]]$

This logical analysis treats the English surface as syntactically ambiguous. The reason is the possible alternative order of the quantifiers, i.e. $\exists x...\forall y$ vs. $\forall y...\exists x$, which have different interpretations.

[7] Such a reinterpretation was the founding assumption of SCG.

[8] The translation mechanism is formalized as (i) a categorial grammar generating a fragment of English, (ii) a homomorphic set-theoretical semantic interpretation, and (iii) the reduction mechanism of typed lambda calculus. Montage uses structural similarity to facilitate the mapping from the syntactic to the semantic representation, while DBS uses it for the interaction between the language and the context level (reference, pragmatics), buffering differences between the two levels (nonliteral use, NLC 5.4.4) by means of inferencing.

In DBS, the alleged ambiguity[9] is not syntactic-semantic, but at best pragmatic.[10] The content is coded as the following unambiguous set of proplets:

13.2.2 DBS ANALYSIS OF A woman loves every man.

$$\begin{bmatrix}\text{woman}\\ \text{cat: snp}\\ \text{sem: indef sg f}\\ \text{fnc: love}\\ \text{prn: 23}\end{bmatrix} \begin{bmatrix}\text{verb: love}\\ \text{cat: decl}\\ \text{sem: pres}\\ \text{arg: woman man}\\ \text{prn: 23}\end{bmatrix} \begin{bmatrix}\text{noun: man}\\ \text{cat: snp}\\ \text{sem: exh pl m}\\ \text{fnc: love}\\ \text{prn: 23}\end{bmatrix}$$

The determiner aspect of the quantifier $\exists y$ representing a(n) is coded in DBS by the sem values indef sg (indefinite singular) of the *woman* proplet, while the determiner aspect of the quantifier $\forall x$ representing every in 13.2.1 is coded by the sem values exh pl (exhaustive plural)[11] and the cat value snp (for singular noun phrase) of the *man* proplet. The verb's cat value decl (for declarative) and its sem value pres (for present tense) complete the picture. The proplets are order-free, and are held together by a common prn value, here 23, which assumes the binding aspect of the logical quantifiers.

In addition to the necessary revisions of the predicate calculus syntax (11.4.6), the reinterpretation of Montague grammar as the hear mode of a talking robot required replacing the semantics of symbolic logic, i.e. definitions in a metalanguage (FoCL Chap. 19), with the recognition and action procedures of a cognitive agent (grounding). To illustrate this point (which has been made from a different angle in NLC Sect. 6.4) let us explain the role of the metalanguage in the truth-conditional interpretation of a maximally simple example:

13.2.3 TRANSLATING Every farmer snores INTO PREDICATE CALCULUS

$$\forall x[\text{farmer}(x) \rightarrow \text{snore}(x)]$$

In Montague grammar, this formula is interpreted relative to a model @ and a variable assignment g, added as superscripts[12] at the end:

---

[9] Montague used traditionally accepted predicate calculus analyses of natural language constructions as much as possible. His main contribution is the quasi-mechanical translation of English surfaces into standard formulas of symbolic logic. The examples analyzed by Montague are comparatively small in number; they are motivated in part by the traditional challenges of analytic philosophy, such as the *de dicto* and *de re* distinction (HBTR Sect. 3.6) and the handling of names (HBTR Sect. 3.4).

[10] See FoCL Sect. 12.5 for the difference between syntactic, semantic, and pragmatic ambiguity. Computational complexity analysis applies only to syntactic ambiguity.

[11] NLC 6.4.7 provides set-theoretical interpretations to the values exh, sel, def, indef, sg, and pl for the purpose of visual pattern matching. See also Choe et al. (2006).

[12] For treating modal operators, Montague defines a model *structure*, consisting of many models and the parameters $i \in I$ (for possible worlds), $j \in J$ (for moments of time), and the variable assignment g. For simplicity, the following example is interpreted relative to a single model @ and a variable assignment g, instead of a model structure.

13.2.4 Interpretation relative to a set-theoretical model

$\forall x[\mathsf{farmer}(x) \rightarrow \mathsf{snore}(x)]^{@,g}$

According to the metalanguage definition of the universal quantifier (Montague 1974, p. 259, (7)), formula 13.2.4 is true with respect to the model @ if the following formula is true with respect to @ and *all* assignments g' to the variable x, usually infinitely many:

13.2.5 Elimination of the quantifier

$[\mathsf{farmer}(x) \rightarrow \mathsf{snore}(x)]^{@,g'}$

Eliminating the quantifier by referring to the variable assignments g' relative to the model @ (defined in the metalanguage) reduces the interpretation of 13.2.3 to the truth table of $[p \rightarrow q]$.

This reduction to the truth tables of propositional calculus may be the crowning achievement of logical semantics for interpreting natural language. For building a talking robot, however, it is a dead end because the machine requires operational implementations instead of definitions in a human metalanguage. Also, the method is not effective because a concrete definition of "the world" as a set-theoretical model is unrealistic.[13]

An analysis of a cognitive agent, natural or artificial, must treat the world as given and concentrate on the agent's interaction with external reality in terms of recognition, reasoning, and action (NLC 1.4.3). The proper functioning of these procedures provides the basis for defining truth as a derived notion.

For example, a talking robot might stumble upon several farmers sleeping off a boozy evening. Based on recognizing this scene, the robot constructs the following content:

13.2.6 Representing Every farmer snores. in DBS

$$\begin{bmatrix}\text{noun: farmer}\\ \text{cat: snp}\\ \text{sem: exh pl}\\ \text{fnc: snore}\\ \text{prn: 27}\end{bmatrix} \begin{bmatrix}\text{verb: snore}\\ \text{cat: \#ns3}'\text{ decl}\\ \text{sem: pres}\\ \text{arg: farmer}\\ \text{prn: 27}\end{bmatrix}$$

At the context level, the core values farmer and snore are provided by the robot's recognition procedures. At the language level, the types of these procedures are re-used as meanings. Instead of two propositions, farmer(x) and

[13] Montague's actual definition of a model $@ =_{def} (A, F)$ is vastly simplified in that it does no more than calling A the domain and F a set of denotation functions. An actual interpretation relative to @, however, would require explicitly listing the elements of A and defining F in a meaningful way, i.e. as modeling "the world." For a deterring example see FoCL Sect. 19.3.

snore(x), connected by the sentential operator $\rightarrow$ of propositional calculus, the DBS content consists of one proposition in which a noun and a verb proplet are connected by address as argument (subject) and functor (predicate).

The content corresponding to Every farmer snores. is true if the robot observes the scene correctly, i.e. if each of the sleeping farmers is in fact snoring. Thus truth in DBS depends on accurate observation and proper reasoning. Moreover, truth – like reference (HBTR 3.1.2) – does not require language. What is required instead is a cognitive agent with agent-internal cognitive content, which may be either of the language or the nonlanguage kind. This is in contradistinction to the ontological assumptions of predicate calculus, for which the presence of a language, natural or artificial, is essential.

13.3 Cognitive Processing at the Now Front

After the failed attempt to program the Montague grammar of SCG because of the computational inefficiency of categorial grammar, CoL tried a semantic interpretation based on semantic hierarchies instead of truth conditions. For building semantic hierarchies (12.4.1) simultaneously with, and driven by, the time-linear NEWCAT syntax (12.3.3), the program underlying CoL used the FRAMEKIT+ program (Carbonell and Josef 1986).

The CoL software was developed until it parsed 421 constructions of English.[14] Long term, however, it was hoped to use the FRAMEKIT+ program also (i) for the storage of content in a frame-theoretic knowledge base, and (ii) for a selective activation of content by navigating along the semantic relations holding between elementary frames. It turned out, however, that the difficulty of pattern matching between frames (12.4.3) extended directly to selective activation, frustrating this renewed attempt towards building the think mode.

Thus, after programming (NEWCAT) and formalizing (TCS) the new algorithm of LA grammar, there was no choice but to design a new data structure and a new database schema. It took decades to evolve a tailor-made solution, consisting in the data structure of proplets and the database schema of a content-addressable word bank – first proposed in FoCL and developed further in AIJ, NLC, and CLaTR'11. The point of departure for designing the word bank database schema was the classic network model (Elmasri and Navathe,

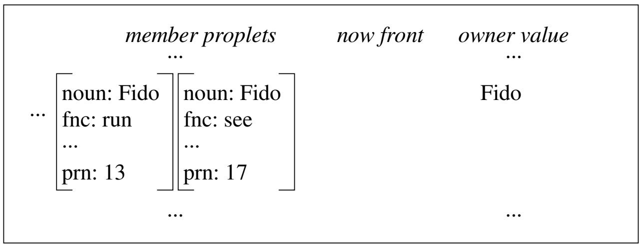
[14] Thanks to Jaime Carbonell for his most generous help in this effort. For several month, he would meet me at his office on Sunday morning to program the start-up software for building the CoL hierarchies with FRAMEKIT+. Written in Franz Lisp, the fresh code was translated by a student into Common Lisp and integrated into the NEWCAT extension the following day.

6th Ed. 2010). The network model had flourished between the late 19 sixties and early 19 eighties, but was superseded by the relational model (RDBMS).

A word bank differs from the classic CODASYL network model in that (i) it uses the data structure of proplets instead of records, (ii) stored data may be read and copied but never changed, the manipulation of data is (iii) limited to the now front (4.1.1) and (iv) based on pattern matching, and (v) any new data are stored in the order of arrival. This may be illustrated with the interpretation of the minimal sentence Fido barked.

Consider the state of a non-empty word bank before new input (proplet of the next word) has arrived, as in the following token line (4.2.1) of *Fido*:

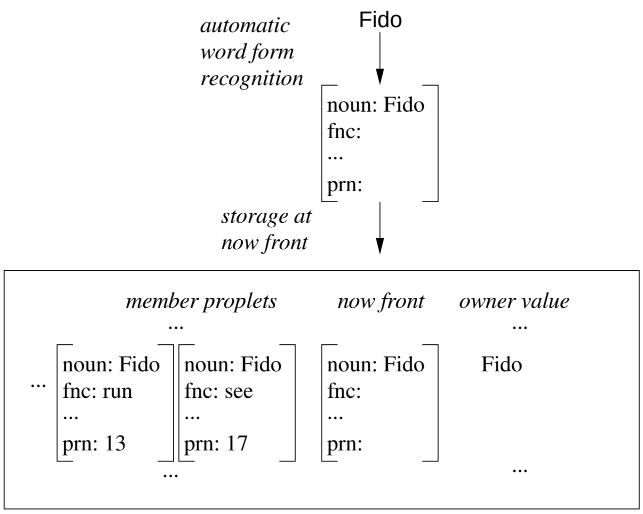
13.3.1 Word bank in state prior to arrival of a next word



The free slot of the now front separates the owner and the member proplets.

When a new word form surface arrives, it is analyzed as a proplet by automatic word form recognition, and stored at the now front in the appropriate token line, here the token line of *Fido*:

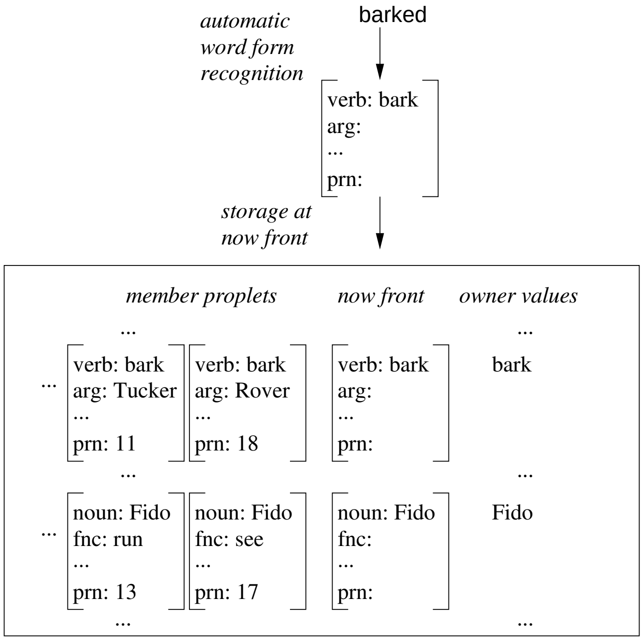
13.3.2 Recognition and storage of the next word



The *Fido* proplet provided by automatic word form recognition to the now front is lexical in that its fnc and prn slots have no value yet.

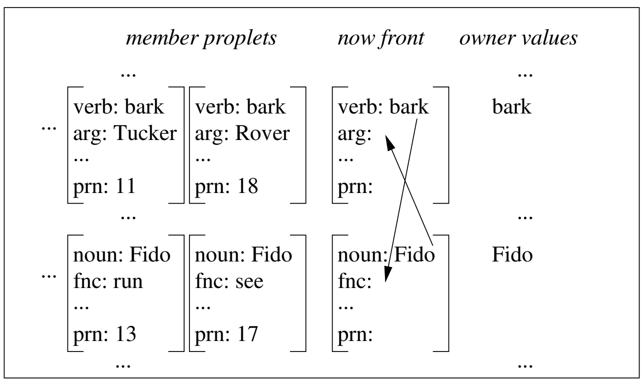
Next another word form surface arrives. It is recognized as a lexical proplet which is also stored at the now front, but in a different token line:

13.3.3 RECOGNITION AND STORAGE OF ANOTHER NEXT WORD



The token lines are ordered alphabetically in accordance with the owner values. To establish the subject/predicate relation, the two proplets at the current now front are concatenated by cross-copying core values into continuation slots:[15]

13.3.4 CONCATENATING *fido* AND *bark* IN SITU



When the punctuation mark is added at the end of a sentence (NLC 13.2.9, 13.3.8, 13.4.10), the now front is cleared of all proplets which have ceased to be possible candidates for any subsequent concatenation (NLC Sect. 11.3). In 13.3.4, this is the *Fido* proplet. The *bark* proplet, in contrast, is taken along by the now front because it may still be needed for the extrapropositional coordination with the verb of a possible next proposition (NLC 13.3.5, 13.4.5):

13.3.5 Result of concatenation and clearing of the now front

		member proplets		*now front*	*owner values*
		...			...
	...	verb: bark arg: Tucker ... prn: 11	verb: bark arg: Rover ... prn: 18	verb: bark arg: Fido ... prn: 23	bark
	...				...
...	noun: Fido fnc: run ... prn: 13	noun: Fido fnc: see ... prn: 17	noun: Fido fnc: bark ... prn: 23		Fido
	...				...

A now front clearing consists in moving the owners of the affected token lines one step to the right, or, equivalently, pushing the affected member proplets one step to the left, as in a push down automaton. The purpose is to make room for the next proposition. Proplets which continue to be active will be left behind as soon as they are no longer needed. In the case of *bark*, this would be after a coordination with the verb of the next proposition (NLC 11.3.6).

The time-linear addition of content to the now front in the corresponding token lines of a word bank has the effect that the proplets left behind in the field of members need not, and may not, be changed. The only way to correct content in a word bank is to write a comment to the now front, referring to the content in question by means of address (Sect. 14.5).

In summary, the results of automatic word form recognition are stored incrementally in the correct token line at the now front and concatenated there. Proplets which have ceased to be candidates for additional concatenations are left behind as member proplets. The loom-like clearing of the now front by moving the owner values one step to the right makes room for the storage and concatenation of the next content. There is no need to sort the just concatenated proplets into the database because automatic word form recognition has placed them already in their final storage position.

[15] As shown by 3.4.1, the assignment of prn values is simple, but left aside in the graphical illustration.

13.4 Selective Activation

The primary key for finding and activating a proplet in a word bank (retrieval) consists of the proplet's core and prn value. The core value allows to access the corresponding token line. The prn value allows to find the wanted proplet in its token line among the member proplets, stored in the order of their arrival.

This form of retrieval may be used for the *selective activation* of content in the form of navigating along the semantic relations between proplets (3.4.3): once a proplet has been activated, a successor proplet may be computed from a continuation and the prn value (spreading activation, Quillian 1968).

Consider the content *Fido barks*, derived in the previous section, stored in the word bank and superseded by later contents:

13.4.1 CONTENT BURIED IN WORD BANK

member proplets		*now front*	*owner values*
...	...		...
... [verb: bark arg: Fido ... prn: 23] ...	... [verb: bark arg: xxx ... prn: yyy]		bark
...	...		...
... [noun: Fido fnc: bark ... prn: 23] ...	... [noun: Fido fnc: zzz ... prn: www]		Fido
...	...		...

The buried *bark* proplet with the prn value 23 may be coactivated by a current *bark* proplet at the now front (NLC 5.4.1). This may start a selective activation along the semantic relations between proplets (5.4.3). For example, the intrapropositional successor is found by using the arg value Fido and the prn value 23. Such an autonomous navigation does not alter the content traversed and may continue as long as continuation values are available. Content activated by a spreading activation will usually differ from the current content by which it was triggered (5.4.1) and may thus alert the agent to potential threats or opportunities even in current situations which would otherwise seem innocuous (HBTR Sect. 4.3).

To utilize a coactivated content, for example, for inferencing, it is "shadowed"[16] to the now front using addresses. Finding a stored content relevant for shadowing is based on automatically turning a current content into a pattern:

13.4.2 Turning a current content into a search pattern

$$\begin{bmatrix}\text{verb: bark}\\\text{arg: Fido}\\\ldots\\\text{prn: 95}\end{bmatrix}\begin{bmatrix}\text{noun: Fido}\\\text{fnc: bark}\\\ldots\\\text{prn: 95}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: bark}\\\text{arg: Fido}\\\ldots\\\text{prn: K}\end{bmatrix}\begin{bmatrix}\text{noun: Fido}\\\text{fnc: bark}\\\ldots\\\text{prn: K}\end{bmatrix}$$

The patterns are derived by replacing the prn value, here 95, with the variable K. The patterns move from right to left through the associated token lines to find all instances of *bark* proplets with the arg value Fido and all instances of *Fido* proplets with the fnc value bark.

13.4.3 Shadowing triggered by patterns at the now front

	member proplets		*now front*	*owner values*
...		...		...
...	verb: bark arg: Fido ... prn: 23	... ... verb: bark arg: xxx ... prn: yyy	verb: bark arg: Fido ... prn: K	bark
...		...		...
...	noun: Fido fnc: bark ... prn: 23	... ... noun: Fido fnc: zzz ... prn: www	noun: Fido fnc: bark ... prn: K	Fido
...		...		...

Once proplets matching the patterns have been found, they are copied to the now front using addresses:

13.4.4 Shadowing a coactivation at the now front

	member proplets		*now front*	*owner values*
...		...		...
...	verb: bark arg: Fido ... prn: 23	... ... verb: bark arg: xxx ... prn: yyy	verb: (bark 23) arg: (Fido 23) ... prn: 96	bark
...		...		...
...	noun: Fido fnc: bark ... prn: 23	... ... noun: Fido fnc: zzz ... prn: www	noun: (Fido 23) fnc: (bark 23) ... prn: 96	Fido
...		...		...

[16] Shadowing differs from "mirroring" as used in computer science. Mirroring creates an exact copy which may stand alone, while shadowing creates addresses which point at the initial referents stored in the word bank (HBTR Sect. 3.5).

The shadowed proposition has the current prn value 96 and refers to the original using the address core and continuation values (bark 23) and (Fido 23).

In summary, first stored content is coactivated by patterns (13.4.2) derived from current content at the now front and extended by completion. Second, the extended coactivated content is shadowed at the now front (13.4.4). Third, the shadowed content participates in the current think mode processing by DBS.Nav and DBS.Inf at the now front. As time moves (13.3.5), the result of these operations will be left behind as sediment as well.

13.5 Inferencing

The inference *If a stranger approaches, Fido barks.* may be applied as follows:

13.5.1 INFERENCE ANTECEDENT MATCHING CURRENT CONTENT

$$
\begin{array}{llllll}
 & \multicolumn{2}{c}{\textit{antecedent}} & & \multicolumn{2}{c}{\textit{consequent}} \\
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: K}\end{bmatrix} & \textbf{caus} & \begin{bmatrix}\text{noun: Fido}\\ \text{fnc: bark}\\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: Fido}\\ \text{prn: K+1}\end{bmatrix} \\
\textit{matching and binding} & \quad\Uparrow & & & \quad\Downarrow & \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: 22}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: 22}\end{bmatrix} & & \begin{bmatrix}\text{noun: Fido}\\ \text{fnc: bark}\\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: Fido}\\ \text{prn: 23}\end{bmatrix} \\
 & \quad\textit{input} & & & \quad\textit{output} &
\end{array}
$$

By matching the input content *stranger approach* with the antecedent, the inference uses the consequent to deduce the output content *fido bark*. The application is an instance of forward chaining (deduction, Sect. 5.2).

Next consider the same inference used in backward chaining (Sect. 5.5), also known as abduction (Peirce, CP 5.171). By matching the input *fido bark* with the consequent, the inference uses the antecedent to deduce *It is likely that a stranger approaches.* In contradistinction to forward chaining, the conclusion here is not a certainty, but a guess at the most likely explanation:

13.5.2 INFERENCE CONSEQUENT MATCHING CURRENT CONTENT

$$
\begin{array}{llllll}
 & \multicolumn{2}{c}{\textit{antecedent}} & & \multicolumn{2}{c}{\textit{consequent}} \\
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: K}\end{bmatrix} & \textbf{caus} & \begin{bmatrix}\text{noun: Fido}\\ \text{fnc: bark}\\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: Fido}\\ \text{prn: K+1}\end{bmatrix} \\
 & \quad\Downarrow & & & \quad\Uparrow & \textit{matching and binding} \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: 24}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: 24}\end{bmatrix} & & \begin{bmatrix}\text{noun: Fido}\\ \text{fnc: bark}\\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: Fido}\\ \text{prn: 23}\end{bmatrix} \\
 & \quad\textit{output} & & & \quad\textit{input} &
\end{array}
$$

The examples 13.5.1 and 13.5.2 differ in the direction of the double arrows. As prn values always increase with additional content, the discrepancy between the K+1 value of the antecedent and of the output derived is interpreted as an abduction by adding the value possible cause to the sem attribute of *approach*.

Next consider an abduction on shadowed content arising from spreading activation. For example, the agent may be replaying events in memory, stumble upon *Fido barked*, and consider the possibility *stranger approached* based on inferencing. The following example uses same the inference as in 13.5.2:

13.5.3 SHADOWED CONTENT MATCHING CONSEQUENT OF INFERENCE

	member proplets		*now front*			*owner values*
1			[verb: approach arg: stranger prn: 89]	⇐	[verb: approach arg: stranger prn: K]	approach
2			[noun: stranger fnc: approach prn: 89]	⇐	[noun: stranger fnc: approach prn: K]	stranger
3	[verb: bark arg: Fido prn: 23]	. . .	[verb: (bark 23) arg: (Fido 23) prn: 88]	⇒	**caus** [verb: bark arg: Fido prn: K+1]	bark
4	[noun: Fido fnc: bark prn: 23]	. . .	[noun: (Fido 23) fnc: (bark 23) prn: 88]	⇒	[noun: Fido fnc: bark prn: K+1]	Fido
	i		ii		iii	iv

Lines 3 and 4 show (i) the original content as member proplets buried in the word bank, (ii) the shadowed content at the now front matching (iii) the consequent patterns of the inference, and (iv) the owner values of the word bank. Compared to 13.5.2, the arrows are rotated from $\Uparrow$ to $\Rightarrow$ and from $\Downarrow$ to $\Leftarrow$. Even though the application order is lines 3 and 4 first and lines 1 and 2 later, the event order is fixed by the prn variables of the antecedent and the consequent of the inference.

Apart from the distinction between current and shadowed input, the application of the inference in 13.5.3 is the same as in 13.5.2. It is just that 13.5.2 shows the application in the standard DBS format of a horizontal matching between the pattern and the content level, while 13.5.3 shows the same matching as it applies vertically at the now front, with alphabetically ordered token lines in the consequent (lines 3 and 4) and in the antecedent (lines 1 and 2). In other words, the overall alphabetical order of the owner values is sacrificed in 13.5.3 to show the inference structure of antecedent and consequent.

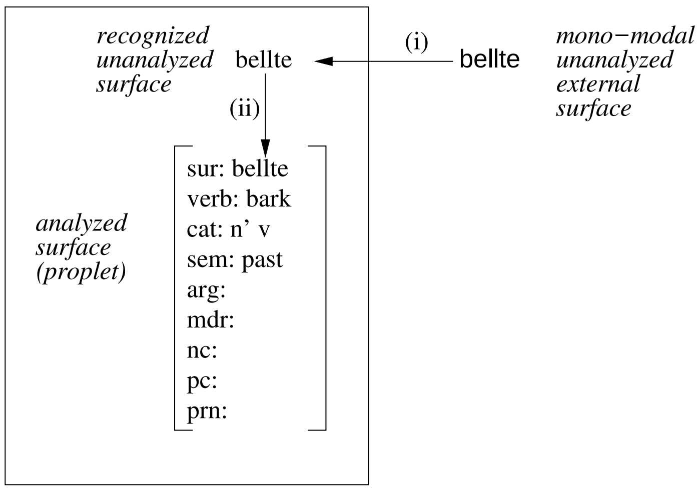
13.6 Language and Nonlanguage Recognition and Action

In DBS, language and nonlanguage contents are coded the same (4.3.3), namely as sets of concatenated proplets. The only difference between language and nonlanguage content is the presence vs. absence of sur values – which are (i) language dependent and (ii) do not participate directly in such processing as hear mode concatenation, think mode navigation and inferencing, or reference.

Thus, what has been shown in Sects. 13.3–13.5 for the cycle of natural language communication holds analogously for nonlanguage recognition, processing, and action (Chapter 8). Changing from language to nonlanguage cognition requires no more than switching the component of automatic word form recognition to nonlanguage recognition, and of automatic word form production to nonlanguage action.

More specifically, language recognition takes a mono-modal modality-dependent unanalyzed external surface, for example German bellte, as input. Retrieval from memory of the associated lexical entry, i.e. the proplet representing the analyzed surface, is based on two steps of pattern matching:

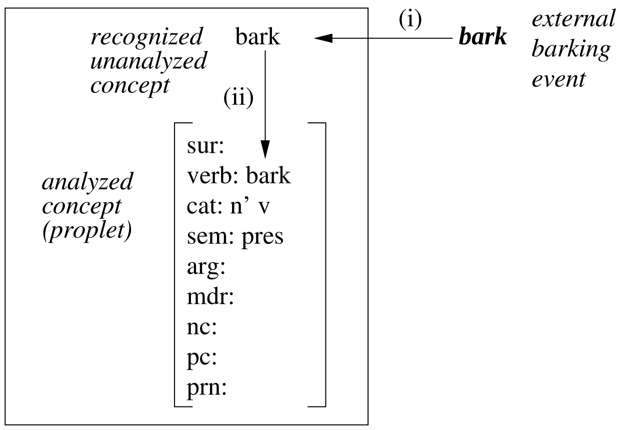
13.6.1 RECOGNITION OF A LANGUAGE SURFACE



The pattern matching takes place (i) between the external surface as raw data and an unanalyzed language-dependent surface type provided by the agent's memory and (ii) between the recognized unanalyzed surface and the corresponding sur value of a lexical proplet (analyzed surface), also provided by memory. Lookup of the lexical proplet resulting from (ii) provides the core and grammatical values for concatenation.

Nonlanguage recognition, in contrast, takes a cluster of unordered multi-modal modality-dependent unanalyzed external features (in the sense of cognitive psychology) as input (8.1.1, 8.1.2, Sect. 8.6; L&I'05; NLC 4.3.1):

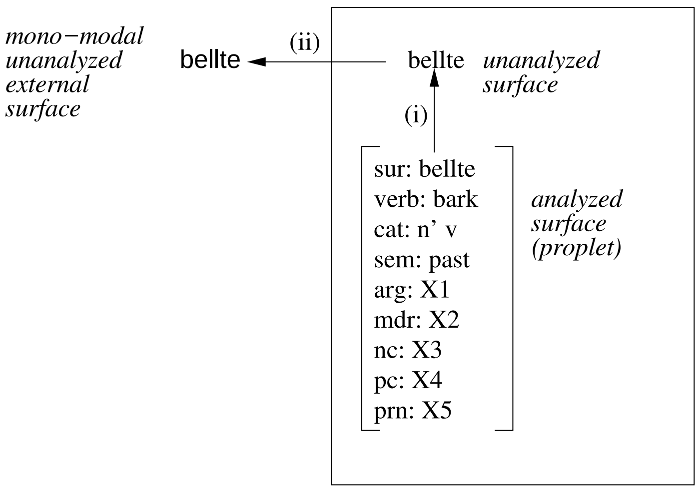
13.6.2 Recognition of a nonlanguage item



The pattern matching takes place (i) between the external barking event as raw data and an unanalyzed concept type provided by the agent's memory and (ii) between the recognized unanalyzed concept and the corresponding core value of a nonlanguage proplet, also provided by memory.

In language production, a lexicalization rule (NLC Sects. 12.4–12.6) derives a language-dependent modality-free proplet-internal sur value, using the language-independent core and grammatical values of the proplet as input:

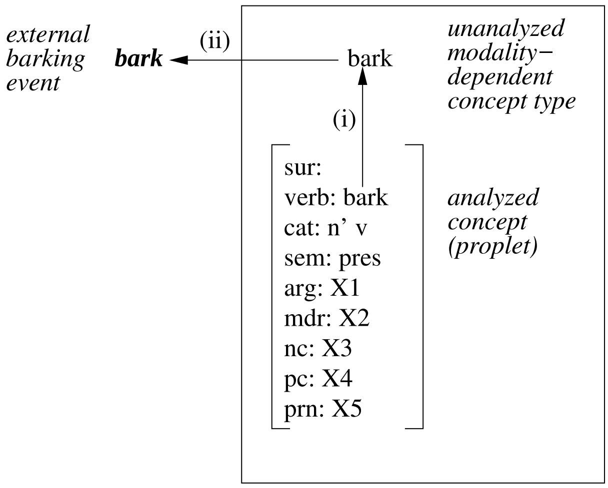
13.6.3 Production of a language surface (action)



The pattern matching (i) transforms the modality-free sur value into a modality-dependent unanalyzed surface type, which is passed to the agent's I/O component (14.3.2, 7). The pattern matching (ii) realizes it as a raw data token.

In nonlanguage action, the proplet's core value, here the concept of barking, is transformed into modality-dependent raw data, here a barking event.

13.6.4 REALIZING OF A NONLANGUAGE ACTION



As in 13.6.2, the sur slot is empty and the sem value is pres (present tense). The pattern matching (i) transforms the modality-free core value bark into a modality-dependent unanalyzed concept type. The pattern matching (ii) realizes it as an unanalyzed external barking event (raw data token) using the agent's action component (NLC 4.4.1).

Language (13.6.3) and nonlanguage (13.6.4) action have in common that the proplets are connected, as indicated by the schematic X1–X5 values, which are obligatory for X1 and X5, but optional for the others. Language (13.6.1) and nonlanguage (13.6.2) recognition, in contrast, are lexical in the sense that their continuation and prn attributes have no values yet.

In summary, the external interfaces are essential for the agent's cognition to monitor and record changes in the external environment by means of recognition, to maintain a state of balance by deriving and applying suitable countermeasures in the form of action, and for communication with language. At the same time, content stored in the agent's word bank may be processed independently of current recognition and action as well as the language/nonlanguage distinction: for reasoning, stored content may be selectively activated, coactivated, shadowed, and inferred upon without involving the external interfaces.

14. Natural Prototype and Artificial Model

Free natural language communication between a human and a talking robot cannot be successful unless certain similarities and compatibilities between the two kinds of agents are ensured. The similarities concern the encoding and processing of language and nonlanguage content in recognition and action.[1] The compatibilities concern the external interfaces of recognition and action.[2]

In other respects, however, the artificial agent may differ from the human model. This affords the opportunity to provide the artificial agent with abilities which substantially simplify, exceed, or complement those of a human. The following sections analyze relations between the natural prototype and technology-based DBS solutions in some specific instances.

14.1 Language Evolving from Nonlanguage Communication

As in the design of a combustion engine, the bottom line in the building of a talking robot is whether it works as intended or not, i.e. whether or not it is able to freely communicate with a human partner in the natural language and the domain of choice. The theoretical foundations for developing the two kinds of machine are quite different, however.

The combustion engine is based on the natural sciences, primarily physics and chemistry, and associated technologies. There is no natural prototype and the interaction with the human user is purely mechanical, e.g. starting the engine by pushing a button and controlling speed by pressing a pedal.

A talking robot also has many mechanical aspects, such as the implementation of artificial locomotion, manipulation, vision, and so on. In contradistinction to a combustion engine, however, there (i) exists a natural prototype which (ii) evolved over 3.7 billion years, and (iii) doubles as the human partner.

Furthermore, regarding natural language communication, the mechanical part is limited to the recognition and production of language surfaces (2.2.1,

[1] Principle of Functional Equivalence (FE, 1.1.1) and the Equivalence Principle for Input/Output (IO, 2.6.3).

[2] Principle of Interface Compatibility (IC, 2.6.2).

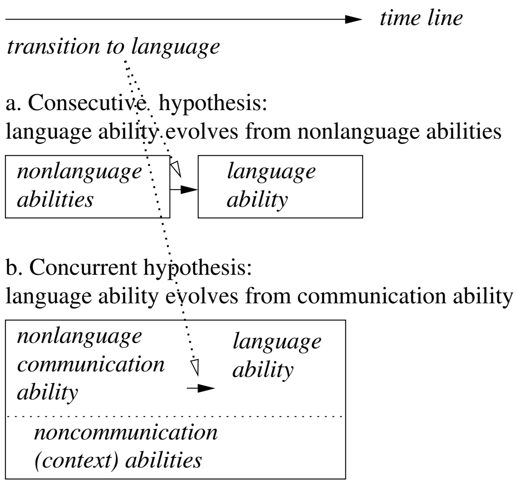
13.6.1, 13.6.3), while the largest part is cognitive: each member of a language community must learn the conventions concerning the lexicon, the rules of the grammar, and the pragmatics of the speak and the hear mode. It is the task of computational linguistics in cooperation with the neighboring fields to reconstruct these conventions equivalently in the software of a talking robot.

The incremental nature of evolution, the late arrival of language in phylo- and ontogenesis, and the complexity of its use in communication lead us to the question from which earlier cognitive component the language ability may have evolved. In light of the functional interaction between the language and the context level in communication (4.3.2), promising candidates are (i) nonlanguage context abilities and (ii) nonlanguage communication abilities.

For example, a chameleon picking up an insect with its long tongue uses a nonlanguage context (noncommunication) ability. Yet, like all animals, chameleons also have an ability to communicate with their conspecifics.[3] As far as we know, the chameleon kind of communication does not satisfy the criteria 3 and 4 listed in 2.2.3 for being a natural language. Instead, the chameleon uses nonlanguage communication abilities, for example, by changing the color of its skin to attract or reject a mate (Karsten et al. 2009).[4]

The dichotomies between language vs. nonlanguage and communication vs. noncommunication may be used to construct two hypotheses about the evolution of natural language, which may be shown graphically as follows:

14.1.1 Two hypotheses: Consecutive vs. Concurrent



[3] In addition to the communication between conspecifics there is the communication between different species, for example, between a plant and an insect for pollination (symbiosis).

[4] A related phenomenon is so-called body language, which is a form of involuntary communication, but does not constitute a language (2.2.4).

According to the consecutive hypothesis, the language ability evolved from nonlanguage abilities. Theoretically, these could be divided into nonlanguage noncommunication and nonlanguage communication abilities. However, because the consecutive hypothesis does not provide for a communication ability, language seems to evolve from noncommunication (context) abilities.

The concurrent hypothesis,[5] in contrast, assumes that the communication and the noncommunication abilities co-evolved from the outset, providing them with the same time to evolve. Thus, when language arrives as the last stage of the communication ability, it can rely on many nonlanguage communication abilities which evolved earlier. Furthermore, because even nonlanguage communication refers,[6] the communication and the noncommunication abilities do not only grow in parallel, but are continuously in close interaction.

For a model of language evolution in natural agents, the concurrent hypothesis is the only choice. This is because the motor driving evolution is reproduction, and communication with conspecifics is at the heart of reproduction – with or without language. That the language ability follows the communication ability, and not vice versa, is shown by (i) language being a refinement of nonlanguage communication, (ii) the late arrival of language, and (iii) the comparatively high complexity of natural language communication.

Building a talking robot, in contrast, is compatible with either hypothesis. This is because artificial agents come into existence by engineering, and not by natural reproduction.

Technically, there is a choice between three overall approaches. The (i) bottom up approach begins with the context level by developing and testing concepts for recognition and action, as well as their concatenation into complex contents; reused as meanings$_1$, these contents are attached to language surfaces by means of conventions. The (ii) top down approach uses language as a window to cognition. It begins with the syntactic-semantic analysis of a language, and re-uses the language structures at the level of context. The (iii) hybrid approach builds non-language and language cognition simultaneously.

On any approach, the software has to account for the close functional interaction and the equal levels of evolutionary development in the communication and noncommunication abilities of a natural agent. The software machine of DBS achieves these desiderata by programming the contents at the language and the context level essentially alike while paying close attention to the inter-

[5] The concurrent hypothesis is in concord with modern work in ethology (M. D. Hauser 1996). It suggests that the communication and the noncommunication abilities of an animal not only evolve in parallel, but their degrees of development are of the same order.

[6] For example, discovering a predator (context recognition) and producing a warning call (communication) constitutes a form of reference.

action with each other and with their external interfaces. Also, by storing language and non-language proplets with the same core value in the same token line[7] (6.4.3), effective storage and retrieval as software concerns take priority over any intuitive distinction between the language and the context level.[8] Re-using software constructs which are designed as standardized, plain, and versatile as possible has always been regarded as good programming practice.

14.2 Agent-Based Spatio-Temporal Orientation

Once the general framework and the associated functions of a DBS robot have been designed in principle, an ability may be realized in various alternative forms. For example, locomotion may be realized naturally as crawling, walking, or running, but also artificially as rolling on wheels, hopping on springs, or flying. If locomotion is implemented in only one form, e.g. rolling on wheels, the agent's concepts for alternative forms of locomotion may be based on the observation of other agents (recognition), but not on self-performance (action).

A striking example of an apparent divergence between an artificial and a natural ability is the spatio-temporal orientation of an agent. The artificial solution has a long tradition in the natural sciences and is based on a Cartesian system of space and time coordinates in which the location of the cognitive agent may be precisely located. Today, this approach may be realized by means of an atomic clock and GPS, which provide highly precise temporal and spatial parameter values, and may be transmitted via radio to the agent, natural or artificial, as continuous updates.

The Cartesian system is an agent-*external* system. Movement is modeled as a point changing location in a three dimensional grid. Time is viewed as changing from one numbered "state-slice" to the next, using natural ($\mathbb{N}$) or integer ($\mathbb{Z}$) numbers for discrete time and rational ($\mathbb{Q}$) or real ($\mathbb{R}$) numbers for dense time. The phenomenon is viewed from a perspective similar to that of a sign-based linguistics and a [-constructive] semantics (FoCL Sect. 20.4).

However, humans have been talking about times and places long before the advent of atomic clocks, GPS, general relativity, and mathematical topology.

[7] Storing corresponding language and context proplets in the same token line greatly facilitates the formal treatment of generalized reference (HBTR 3.1.2).

[8] This in contradistinction to tendencies in language analysis which aim to separate language and thought, as witnessed, for example, by Chomsky's "autonomy of syntax," and Frege's (1879) attempt to construct a graphical code of "pure thought." Respectfully but mistakenly we tried to accommodate this idea in NLC (2006) by using separate, but alternating LA grammars for the think and the speak mode. This caused a host of technical difficulties in programming which magically disappeared when the speak mode was integrated into the think mode by defining language-dependent lexicalization rules which are inserted into the sur slot of the proplets traversed by DBS.Nav.

The Cartesian system is therefore a welcome addition to getting oriented in time and space, but it is not the primary method of natural agents.

What would be a natural solution for a cognitive coding of time and space? In DBS, the low-tech method of spatio-temporal orientation is based on an agent-*internal* fix point. This is the now front (4.1.1, Sects. 13.3–13.5) as the place where the incoming (recognition) and outgoing (action) data are processed.[9] Among these are spatio-temporal *landmarks* which relate the agent's cognition to external locations and points in time.

When such a landmark content is stored at the now front, it is quickly superseded by subsequent contents (14.3.2). Yet the arrival (recognition) and departure (action) order is preserved by their position in the token lines of the agent's memory, similar to positions in layers of sediment. This mechanism is part of a DBS robot's basic cognitive layout, and nothing more is needed for naturally keeping track of the agent's location and the passing of time.

The content of temporal landmarks is characterized by values which are often cyclical, like the change of seasons, the phases of the moon, and the alternation between day and night. There are also noncyclical temporal landmarks, for example, the birth of a child, as in B.C. and A.D. Spatial landmarks may be values characteristic of a location, like a large rock or an unusual tree.

In modern times, the natural landmarks have been supplemented by artificial landmarks such as church bells, clocks, watches, maps, and sign posts. Their use by cognitive agents does not differ from the use of natural landmarks. Whether a natural or a low tech landmark is sufficient or a high tech landmark is needed depends on the task.

For example, the agent usually does not require GPS for responding adequately to Go to the bakery!. Instead it may be sufficient to remember a habitual walking procedure. Thereby, a low tech landmark like a sign post with a street name or a shop sign may be helpful and be used in the same way as a natural landmark.

Based on recognition, coactivation, and inferencing, a DBS robot continually updates its current spatio-temporal position in the manner described. This produces more or less precise values for the S and T attributes of the agent's STAR-0 (Sect. 10.1). The ST values (i) provide the agent with current spatio-temporal orientation, (ii) help to anchor content in the agent's continuous monitoring, (iii) serve as the referents of spatio-temporal indexicals in language signs, and (iv) provide the basis for computing the speaker's perspective on

[9] It is also the place from where the spatio-temporal indexicals (FoCL Sect. 6.3) such as there or tomorrow are computed on the fly (Chapt. 10: STAR).

stored content (STAR-1, Sect. 10.2) as well as (v) the hearer's perspective on language content (STAR-2, Sect. 10.3).

14.3 Functional Flow from Recognition to Action

In the construction of a DBS robot, the interdependency between the agent's capabilities and the agent's ecological niche allows the designers to control two different, but interacting systems: the (i) artificial agent and its (ii) environment.[10] For example, by keeping the environment constant, more and more successful agents may be designed for it. Conversely, a given agent may be tested in more and more challenging environments.

Such two-pronged incremental upscaling emulates evolution, though in much shorter time spans. Also, the development is carried out by hard- and software engineers who design the agents and their task environments, while evolution proceeds naturally based on reproduction and adaptation.

The interaction between an agent, natural or artificial, and its environment is driven by the principle of balance (5.1.2). Deviations from a current state of balance activate the agent's autonomous control to derive and execute appropriate countermeasures. Thereby, recognition, coactivation with completion, inferencing, and action are integrated incrementally as follows:

14.3.1 Incremental integration of different procedures

1. Current recognition provides an initial language or context proplet. It is lexical and stored at the now-front (4.1.1, 13.3.2) at moment t_1, thereby subactivating the associated token line (5.4.1, first step of coactivation).
2. Current recognition provides a second lexical proplet (13.3.3). It is also stored at the current now-front and also subactivates a token line.
3. The rule component concatenates the two proplets *in situ* (13.3.4) into a new sentence start. Replacing the prn value of this sentence start by a variable (13.4.3) results in a pattern which is used for intersecting the two token lines (5.4.2, second step of coactivation).[11]
4. By adding a new lexical proplet to the resulting sentence start, this incremental cycle is repeated, (i) extending a propositional content and (ii)

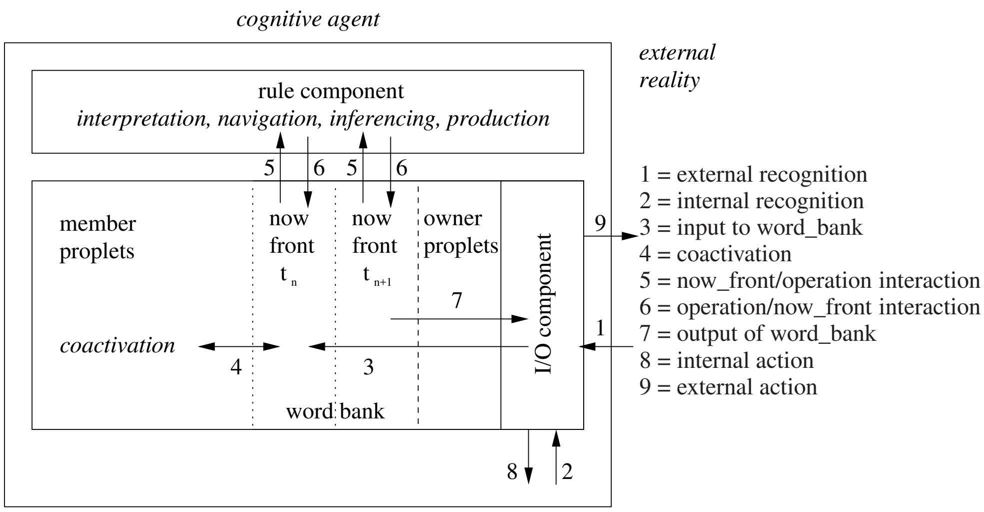
[10] The environment may be natural or artificial, but it must be real in the sense that recognition and action are performed autonomously by the agent's interfaces, instead of being simulated as in VR.

coactivating and shadowing older contents (13.4.2) at the now front. Coactivation may be expanded by completion of various degree using DBS.Nav (5.4.3) for spreading activation.

5. Current and shadowed contents are used as input to inferences (Sects. 5.2, 5.5; 13.5.1–13.5.3), the output of which is written to the now front.
6. The outcomes of inferencing are evaluated relative to (i) the agent's current situation, represented at the current now front, and (ii) past experiences, as coactivated and completed sets of member proplets shadowed at the now front.
7. The inference with the best outcome is chosen as a blue print for possible action and written to the now front at the next current moment t_{n+1}. If an action has been selected, the blue print is passed to the I/O component for realization (Sect. 5.6). If there is no clear favorite, there may be a random choice (trial and error, Sect. 6.3).

By locating the central processing of cognition at the current now front, the two main modes of (i) recognition (interpretation) and (ii) action (inferencing) may be incrementally integrated into the time-linear processing cycle based on arrival order. This cycle may be shown graphically as follows (repeated from 4.5.4):

14.3.2 Time moving through the agent's now front



The now front at moment t_n shows the interpretation of an input, the now front at moment t_{n+1} shows the application of an inference deriving a blue print for action. The right hand side of a current now front is delineated by the column of owner values; the left hand side fades into the permanent sediment of the word bank.

For the agent, the now front is stationary, but the data move continuously through it in time, leaving stored content as sediment behind (loom-like clearance). More specifically, each time a now front slot is filled by a newly arrived proplet, the slot is reopened by moving the affected owner values one step to the right, or equivalently by moving the affected member proplets of the token line one step to the left, as in a push-down automaton.

As the current now front stage t_n is superseded by a new stage t_{n+1}, t_n proplets which have ceased to be candidates for further concatenation are left behind to join the sediment of the member proplets. Proplets which are still potential candidates, however, are excepted, remaining available at the now front as long as necessary (Sect. 13.3; NLC Sect. 11.2).

The content at the now front is used for continuous coactivation[12] (Sect. 5.4). This requires massive, multiple search in real time. The quality of individual search operations depends on (i) the speed of the retrieval mechanism and (ii) sufficient expressive power enabling queries to retrieve content at the appropriate level of detail. In a DBS system, speed is provided by the database schema of a content-addressable word bank and the use of pointers. Expressive power is provided by query patterns which use the same semantic relations (3.2.3, 3.2.6) and some of the same constant values (4.2.2) as the content proplets searched for.

An alert DBS robot may require running several tasks in parallel, for example recognition, coactivation, shadowing, and inferencing. For synchronizing such competing tasks,[13] DBS coordinates the steps of each task as follows: if there are *k* parallel strands, step n of all *k* strands is completed before moving on to step n+1 of all *k* strands. Technically, the synchronized parallel steps of these strands may be executed in parallel (using multiple processors, hardware solution) or sequentially (using the operating system, as in UNIX scheduling, software solution). The advantage of the software solution is that there is no fixed limit on the number of parallel strands.[14]

[12] To overcome the inflexibility of contemporary systems. Mohammadzadeh et al. (2005) propose "template guided association," aimed at XML. DBS coactivation is similar: the templates are the concatenated pattern proplets. In addition to a highly effective primary key, defined as an address implemented as a pointer, DBS provides the option to search for continuation values, morphosyntactic categories, base forms, matching inferences, memorable outcomes, n-grams, frequencies, and so on.

[13] See Herlihy and Shavit (2011) for related issues.

14.4 Routine as a Chain of Subactions

The procedures of (i) taking a surface as input and deriving a content as output in the hear mode and of (ii) taking a content as input and deriving a surface as output in the speak mode are being hooked-up in the DBS laboratory set-up:

14.4.1 DBS LINGUISTIC LABORATORY SET-UP

$$\begin{array}{ccccc} & \textit{hear mode} & & \textit{speak mode} & \\ \text{surface\_x} & \Longrightarrow & \text{content} & \Longrightarrow & \text{surface\_y} \\ & & \Updownarrow & & \\ & & \textit{think mode} & & \end{array}$$

The laboratory set-up is special because (i) the hear mode precedes the speak mode in one and the same agent, (ii) the content derived in the hear mode is reused in the speak mode, and (iii) surface_x = surface_y.

In communication, in contrast, a speech act (i) begins with the speak mode of agent A and ends with the hear mode of agent B (2.3.1, 2.6.1). Also, when an agent switches from the hear mode into the speak mode, (ii) the content to be realized in language will normally not be the same as the content just interpreted, and (iii) surface_x≠surface_y.

The purpose of the DBS laboratory set-up is to ensure that the derivation of content in the hear mode provides all the details needed for a corresponding speak mode. Once the software handles a certain test sample of the language in question, the restrictions of the laboratory set-up are replaced by an autonomous control (*think mode*) with the purpose of maintaining the agent in a state of balance vis à vis a changing environment. Then there will be two agents, the speak mode of one will precede the hear mode of the other, and the content just interpreted will differ from the content produced when an agent switches from the hear to the speak mode.

Maintaining balance applies not only to the correction of an imbalance, such as eating as a countermeasure for hunger (Sect. 5.5), but also to preventing an imbalance from occurring in the first place, such as drinking from a cup without spilling. In either case, the action may be analyzed as a time-linear sequence of subactions, as in the following routine:[15]

[14] If the current maximum were suddenly exceeded, the hardware solution would have to physically add another processor in order to maintain the level of efficiency, whereas the software could be easily programmed in advance to be adaptive. Of course, every additional processor helps.

[15] In early A.I. such a task was formulated in terms of theorem proving (Fikes and Nilsson 1971).

14.4.2 Subactions of the "drinking from a cup" routine

pick up cup
raise to mouth
drink
lower cup
set cup down

In concord with the Continuity Condition (7.2.1), the routine is implemented as a chain of inferences, i.e. *pick up cup* ⇒ *raise to mouth, raise to mouth* ⇒ *drink, drink* ⇒ *lower cup, lower cup* ⇒ *set cup down*. In short, the format of inferencing used for routines resembles the format of inferencing used for problem solving as illustrated in Sect. 5.2.

Routines may be established by observation or by schooling. For example, learning how to drive will involve the following subactions:

14.4.3 Subactions of starting a car from the curb

start engine
check traffic
move out of slot
enter flow of traffic

The subactions making up everyday routines have to be in a certain logical order and must be performed in harmony.[16] If they are subsymbolic (Tilden and Hasslacher 1994), an artificial agent may acquire them by means of the guided pattern method (Sect. 6.2). Communication with language may also be helpful for maintaining balance, as in the request Always lock this door at night!

The similarity of inferencing used for performing routines and for problem solving provides for flexible behavior. In novel situations, subactions available from a routine may be selected and reordered for the purpose of problem solving. The goal is to find a chain of subactions from a state of imbalance to a solution, i.e. regaining a state of balance. This may involve trial and error.

14.5 Data Correction in a Word Bank

Once a piece of content has been stored in a content-addressable word bank, it may not be changed or manipulated in any way. The reasons are as follows.

[16] In today's robotics, this involves self-collision detection, path planning, obstacle avoidance, and wide task availability.

First, revisions of content would be impractical because maintaining consistency would require huge amounts of update work.[17] Second, any change of the data would diminish or destroy their value for the user because they would cease to be trustworthy. Third, revisions or corrections of content are *not needed*.

The one and only way of correcting data in a word bank is writing a comment to the now front, like a diary entry.[18] For this, it is sufficient that stored data may be read and copied by address as often as desired, and be accessed from any query angle without touching the originals. Using addresses to refer to stored data[19] allows to involve an original referent in as many semantic relations of structure as needed (4.4.3). Comments referring to stored content are the result of inferences which take current or shadowed content as input, and write their output to the current now front (14.3.2). In this way, a historically accurate record of a comment's origin is preserved for later use, by the same, automatic means as any other freshly derived content (Sect. 13.3).

There are different kinds of comment. The kind which serves a similar function as data correction in conventional databases is the *correcting comment*. It says that something asserted in the past was not so and how it really was. This may apply to a faulty instruction, a recall, or a (re)evaluation of historical events – like the different views in the years 1938 and 1959 of whether Joseph Stalin was good or evil. The version currently held true by the agent is the most recent, i.e. stored rightmost in the token lines.

Another kind is the *expansion comment*, for example, supplying the facts of a town to the town's name, as in a dictionary. There is the *funny comment*, as in a caricature of Nixon and Brezhnev. Also, there are the *emotional comments*, the denials and inventions, the prejudices, the comments applying to a specific domain or to a specific event, and so on.

The inferencing producing comments is driven by current input which coactivates stored data (Sect. 5.4). When spreading activation lights up a content, any and all addresses pointing at it may be subactivated as well. When subactivation lights up an address, the original and all the other addresses pointing at it may also be subactivated. In this way, the original referents and their

[17] Take, for example, the futile effort of rewriting an event in political history comprehensively, consistently, and convincingly every time when the winds of changes shift.

[18] Restricting corrections to comments effectively curtails fraud: the sanctity of the data may be much more easily ensured technically if any data modification, for whatever reason, is excluded in principle.

[19] The database may register visits to a content by keeping a "visitors list" in the proplets traversed. Using addresses, such counters are associated with specific contents, but do not change them. Monitoring visiting traffic provides the basis for valuable statistical information regarding the agent's cognitive operations.

coreferent copies may be systematically coactivated, resulting in a spreading activation which may increase the power of reasoning.

By accompanying the artificial agent's current monitoring continuously with corresponding data in memory, coactivation resembles the natural prototype, at least functionally. Technologically, however, coactivation in a DBS robot differs from brain activation. In brain scans, activation lights up *meaningful areas*. This has no analog in an active word bank because proplets are organized in (i) the alphabetical order induced by their core value and (ii) their temporal arrival order. Thus if we view the word bank as a large field of many alphabetically ordered token lines, the lighting up in connection with the coactivation of a certain content would appear as a seemingly random distribution all over the field of member proplets.[20]

14.6 Avoiding Memory Overflow

Stored content which never changes makes practically no processing demands on the system. The personal history contained in an agent's static sediment of memory provides an individual notion of what really happened, what is true, and what is right. There is also a natural definition of the agent's current state: it consists of the most recent data, coming in at the now front and stored last (rightmost) in the token lines.

These are important advantages, but a continuous massive stream of input data never to be touched threatens memory overflow. How does nature solve this problem? Roughly speaking, nature uses (i) reproduction to pass general data from one generation to the next via gene memory and (ii) a finite life cycle to delete all personal knowledge and experience by destroying the hardware of the individuals' brain memory (death). Thus, each individual has to learn from scratch how to walk and talk, and to experience the drama of how to survive and to reproduce.[21]

In some special instances, personal knowledge may be partially preserved in the form of agent-external *witnessing remnants* (exograms, Donald 1991), such as ancient cave paintings, pottery, clay tablets, papyri, and so on. These require careful, highly informed interpretation, just as the light from distant

[20] The search keys of the natural memory in humans seem to be shapes such as syllable structure. For example, when looking for the name of the person who composed the music of *Lulu* and *Wozzek*, memory may bring up the correct photography of a distinctive face and the name Cesar Frank. The latter is clearly not correct, but shares a syllable structure with Alban Berg.

[21] Individual experiences may be transferred to gene memory by epigenetic mechanisms currently under intense study. By these mechanisms, information regarding individual behavior which turned out to be successful, but also personal trauma, may enter the data pool transmitted by the reproductive chain.

galaxies in astronomy, sediment in geology, and fossilized bones in paleontology. The interpretation of witnessing remnants may fill up an individual agent's personal memory in the same way as day-to-day events like the experience of an outstanding musical performance.

The filling up of a personal memory may be decelerated by filtering and cleaning up the incoming data prior to storage. Following the natural prototype, this may be done by providing a short-term and a long-term memory, and by doing selection and cleanup in short-term memory before long-term storage. For example, subjunctive and imperative action sequences (Sect. 5.6) may be derived in short-term memory, but often only the imperative sequences need to be remembered long-term.

The other method to stave off memory overflow, also following the natural prototype, is forgetting. Though a limiting case of data change (and therefore not really permitted in a content-addressable word bank), forgetting is functionally acceptable if the items deleted do not result in any, or result in only a few, dead ends in the associative data network. As a solution, Anderson (1983) proposed a frequency-based approach, namely to bleach those data from memory which are never or very rarely activated by the cognitive operations of the agent.[22] The procedure resembles garbage collection in programming languages like LISP, Java, or Oberon.

In a word bank, the many dispersed little spaces created by bleaching may be made re-usable by pushing the proplets in a token line to the left, like pearls on a string (defragmentation). In this way, the ordering of proplets in a token line is preserved, while the contiguous spaces reclaimed on the right-hand end become available for storing new data.

The declarative addresses, consisting of continuation and prn values, which code inter-proplet relations (Sect. 4.4) provide a straightforward solution to the recomputation of the pointers' to new physical storage locations, necessary after bleaching and defragmentation. The periods in which bleaching, defragmentation, and recomputation are performed in artificial agents may be likened to the periods of sleep in natural agents.

A managing of the data stream inspired by natural phenomena such as sleep and forgetting may be complemented by purely technical means. Just as an aircraft may be designed to maximize payload, range, speed, etc. with little or no resemblance to the natural prototypes (e.g. birds), there may be a software

[22] To handle the sudden coming back of long forgotten details, as in Proust's Madeleine episode, the intensity of a memory in its historical context must be taken into account. In general, long-term memory seems to have a life of its own. As Cormac McCarthy (2006) puts it:

> You forget what you want to remember
> and remember what you want to forget.

routine which is applied whenever the amount of memory in the word bank reaches a certain limit; the procedure compresses the currently oldest content segment (containing, for example, all proplets with prn values between 30 000 and 32 000) and moves it into secondary storage without any deletion (and thus without any need to choose).[23]

[23] Depending on the application, more differentiated routines may be designed.
The only downside to the data in secondary storage is a slight slowdown in initial activation. If secondary storage is not practical, as on a deep space mission, for example, a robot's primary memory must be sized to accommodate the amount of data incurring in the expected lifetime. In robots accessible for maintenance, the hardware for primary memory may be expanded whenever needed.

15. Corpus Linguistics

The overall processing time in the hear and the speak mode depends on the whole cycle of natural language communication, including the morphological, syntactic, semantic, and pragmatic details of the language at hand. Discovering these details and programming them for uploading into the DBS software machine may take years and is open-ended because of the continuous evolution of living natural language and its supporting culture.

The upscaling process must maintain real time performance from the beginning. In speech, the hearer's interpretation should be completed with the end of the current sentence to avoid lagging behind in the next. The speed of speaker's production should strive for optimal intelligibility, whereby the upper limit on velocity is the capability of articulation. In writing, in contrast, the speed of interpretation is controlled by the hearer/reader and may take as long as it takes; the speed of production depends not only on the writer's skill to produce letter sequences, but also on the time it takes to formulate.

15.1 Four Levels of Abstraction

For the computational reconstruction of the language-dependent details, the examples may be in either of the two main modalities: for communication, the heuristic procedure proceeds from analyzing written language to producing and interpreting spoken language, but (ii) for verification and correction, it may proceed from producing and interpreting spoken language to a linguistic analysis as written language.

Written text differs from other online data such as photographs, videos, spoken language, or music in that it consists primarily of letters. In the electronic medium, written text may be represented at the following levels of abstraction:

15.1.1 The four levels of abstraction for representing text

– Level-one: *Representation as bitmap*
  Pages are scanned into the computer as bitmaps. These preserve the appear-

ance of the page (which may be important, as in a medieval manuscript), but do not allow any letter-based text processing.

- Level-two: *Digital representation*
  The bitmap representation is transferred automatically to a digital representation (e.g. ASCII or Unicode) by means of an OCR software or the words are typed directly into the computer (Sect. 2.3). The digital representation allows text processing, such as (i) automatic search for, (ii) simultaneous substitution of, and (iii) movement or deletion of words or phrases, based on alphanumeric sequences and an index (inverted file).
- Level-three: *Representation with metadata markup*
  The digital letter sequences are enriched with manual markup, for example in XML, which characterizes chapter and/or section headings, the paragraph structure, name of the author, bibliography, etc., depending on the kind of text, e.g. newspaper article, novel, play, or dictionary. This allows to change from one print style to another while maintaining the text structure as encoded by the markup.[1] Furthermore, the markup may be extended to a classification of content, for example the text's domain, thus supporting retrieval.[2]
- Level-four: *Representation as content*
  The text is represented as content. The content is automatically derived from the letter sequence by means of a rule-based syntactic-semantic parser. The resulting output depends on the underlying linguistic theory.

The retrieval power of search engines such as Google or Yahoo is based on matching alphanumeric level-two sequences in an online text, regardless of whether they happen to represent a frequent or an infrequent word form, a neologism, an acronym, an expression of a foreign language, or simply nonsense.

For the same reason, level-two letter sequences do not provide any grammatical information. For example, swimming and swam are not recognized as forms of the same word, i.e. *swim*. Furthermore, the grammatical distinctions between noun, verb, and adjective, singular and plural, the syntactic and verbal moods, the tenses, etc., are not specified. Without them, a grammatical analysis is not possible.

[1] This kind of metadata markup evolved from the manual annotation of manuscripts in print shops.
[2] This kind of metadata markup originated in the manual cataloging work of the library/information sciences.

15.2 Statistical Tagging and Metadata Markup

In natural language processing (NLP), the first attempt at introducing grammatical distinctions was TAGGIT by Francis (1980), a pattern-based categorization software which required much post-editing. Building from there, Garside, Leech, et al. (1987) developed the CLAWS1 system. It tries to induce the categorization from the statistical distribution of word forms in texts. This *tagging* was developed in part for getting retrieval results from large corpora which are better than letter-based level-two pattern matching alone.

Tagging is based on a level-three markup (15.1.1) of a small part of a corpus, called the *core corpus*. After hand-tagging the core corpus, the probabilities of the transitions from one word form to the next are computed, usually by means of *Hidden Markov Models* (HMM).[3] Then the transition probabilities of the hand-tagged core corpus are transferred to the whole corpus. Finally, the tagged corpus is post-edited.

This process requires language data fixed in storage. Therefore it cannot run in real time, making it unsuitable for spontaneous speech in free human-machine dialog. Also, the results produced by the statistical tagger are not good enough for syntactic-semantic parsing because of a high error rate.[4] The remedy is massive manual post-editing: it has to be done for every corpus from scratch and introduces human error (inconsistent post-editing).

An alternative to the manual post-editing of statistical tagging is the manual markup of the original text. Free from the struggle to optimize the statistical part of the tagging process, manual markup takes pride in developing *standards* for systematic annotation. As a simple example, consider the date specified in 11.1.1 as Jan. 16th, 1832 and a standardized[5] annotation in XML:

15.2.1 Standardized annotation of a date in XML

```
<startdate>1832-01-16</startdate>
```

Such annotation may help to improve precision, recall, and speed for a computational retrieval of dates from textual databases.[6]

---

[3] The use of HMMs for the grammatical tagging of corpora is described in, e.g. Leech, Garside and Atwell (1983), Marshall (1983), DeRose (1988), Sharman (1990), Brown, Della Pietra, et al. (1991). See also Church and Mercer (1993).

[4] FoCL 15.5.3. See also NLC, Introduction VI.

[5] The format may be chosen from numerous XML standards: the Library of Congress, May 05.2015 lists 21 MARC Formats and Fields standards, creating an embarrassment of riches (FoCL Sect. 9.5).

[6] In order not to clutter up the original text, any annotation should be added in *stand-off* (Ide et al. 2010) i.e. visible for computers but leaving the appearance unchanged for humans. A stand-off annotation

Like the manual postediting of statistical tagging, manual annotation is a substantial source of human error because the standards are complicated even for experts. Also like statistical tagging, manual markup is limited to language fixed in storage (i.e. it does not run in real time) and has to be done from scratch for each text.

The DBS alternative is automatic word form recognition (FoCL Chaps. 13–15) and syntactic-semantic parsing as developed for the talking robot's hear mode (NLC Chaps. 11 and 13): instead of relying on annotation, characteristic items in a text, such as dates, may be retrieved by patterns. For recognizing a new kind of characteristic items, new patterns may be easily integrated into the DBS system, resulting in a permanent extension of on-the-fly parsing.

15.3 RMD Corpus

The core values, the semantic relations, and the levels of grammatical complexity are agent-internal constructs of cognition. In a talking agent, their processing must be in real time. Recorded language data, in contrast, are agent-external objects with the advantage that their analysis may take as long as it takes, i.e. it does not have to be in real time.

A classic example of recorded data collected for linguistic analysis is the Brown Corpus (Kučera and Francis 1967, Francis and Kučera 1982), designed as an electronically stored monolingual synchronic corpus. Its scientific purpose is to provide an accurate snapshot of American English in the year 1961. A well-built corpus contains the vocabulary, constructions, collocations, idioms, and frequency distributions characteristic of different domains (genres) in the language and the time interval selected. Known frequency distributions may help to optimize the recognition rates for word form tokens and similarly for the parsing of syntactic-semantic constructions.

According to Kučera and Francis (1967), a well-built[7] corpus should be *representative* and *balanced* (FoCL Sect. 15.3). However, because no fixed, one-shot corpus like the BNC can be *proven*[8] to fulfill these desiderata (Oostdijk 1988), an empirically more broad-based, more differentiated, more interest-

---

is based on an *inverted file*, i.e. an index which allows to find letter sequences in text. The index is constructed automatically (Tompa 1992) by (i) numbering the letters from beginning to end of the text and (ii) constructing the inverted file by listing for each letter of the alphabet the position numbers in the text, e.g. a: 23, 34, 52.... The resulting index has a size similar to the text itself (FoCL Sect. 2.1). If there is any change in the text, the position numbers become invalid and the inverted file has to be computed anew. By connecting word forms to the positions of their letters in the text, the latter may by used for retrieving the former. The off-line annotations are written into a separate file and related to positions specified by the index.

ing, and more long-term approach is the construction of a standardized RMD corpus, i.e. a Reference-Monitor corpus structured into Domains.

The reference corpus is constructed from numerous subcorpora for different domains, such as anthropology, architecture, astronomy, biology, chemistry, ecology, entomology, ethology, everyday language, fiction, history, law, medicine, music, philosophy, physics, politics, religion, and sport (von der Grün 1998). Establishing the domains may be based on the cataloging work of the library/information sciences. The sizes of the domains may be determined by methods which evolved from those used for the Brown corpus.

Once the reference corpus has been constructed for the initial year, it is followed by annual monitor corpora (Sinclair 1991). They must resemble their reference corpus in every way: overall size, choice of domains, domain sizes, etc. This may be easily achieved by building the reference corpus and the monitor corpora with texts from the same carefully selected basket of *renewable* language: long-running newspapers for everyday language, established journals for specific domains, and a selection of fiction from the year in question.

Each annual monitor corpus requires linguistic analysis. Questions of quality aside, statistical tagging is not suitable for this because of its need for massive manual postediting, which may take more than a year[9] (Sect. 15.2). The DBS alternative is automatic processing in real time. This applies to (i) collecting the texts once the initial set of renewable sources has been settled on, (ii) the statistical analysis once a useful routine as been established, (iii) automatic word form recognition, (iv) syntactic-semantic parsing, (v) storage in a DBS corpus database (Sect. 15.5), and (vi) comparison with preceding monitor corpora. Automatic processing ensures the quality of standardization necessary for meaningful comparisons between monitor corpora. It also avoids instructing, supervising, and remunerating legions of listless low-wage laborers.[10]

An annual succession of monitor corpora allows a detailed view of how the language and the culture are developing, in different domains and over many decades. Statistical analysis will show, for example, how politics and natural

[7] A well-built corpus is in contradistinction to some random, all-you-can-get collection of texts, called an “opportunistic corpus” and offered as a free “resource” waiting to be used.

[8] A complication of a word form’s frequency ranking are asymmetries between the speak and the hear mode. For example, a newspaper article or a radio address is produced once, but interpreted by many. Conversely in an uprising: many people may chant the same demand to a single dictator.

[9] The manual postediting of the BNC tagging took more than a decade (Burnard ed. 1995, 2007). However, an RMD corpus may be built even today as long as the manual postediting can be done within a year, i.e. before the arrival of the next monitor corpus.

[10] In addition to the difficulty of achieving consistency of manual markup within a project, there is the difficulty of different projects using different markup conventions. Thus comparing them requires a means of translating between different annotations. This has resulted in the budding field of “interoperability” in the manual markup business.

disasters cause a temporary frequency increase of certain words in certain domains. A carefully built RMD corpus is in the interest of the whole language community and should be entrusted to the care of a national academy.

15.4 The Infinitive

The quality of automatic corpus analysis depends on the quality of the linguistic theory regarding completeness of function and of data coverage. Assuming for the sake of the argument that DBS does not yet handle the infinitive (NLC 6.6.5–6.6.7), it would not be available for corpus processing. Let us therefore consider how to fill this gap for English.[11]

Following the grammars for classical Latin, traditional grammars of English[12] call the unmarked form of the verb the infinitive, classify it as non-finite and uninflected for person, number, and tense, and use it as the verb's base form (citation form). From a surface compositional point of view it must not be overlooked, however, that the surfaces of such infinitives as (to) sleep, (to) see, and (to) give are the same as the finite non-third-person singular present tense forms in (I) sleep, (you) see, and (we, they) give, respectively.[13]

This is reflected in the following DBS hear mode derivation of John tried to read a book, showing a transitive infinitive serving as the object[14] of try:

15.4.1 HEAR MODE DERIVATION OF John tried to read a book

lexical lookup

	John	tried	to	read	a	book	.
	noun: John cat: snp fnc: mdr: prn:	verb: try cat: n' a' v arg: mdr: prn:	verb: v_1 cat: inf fnc: arg: prn:	verb: read cat: n−s3' a' v arg: mdr: prn:	noun: n_1 cat: sn' snp fnc: mdr: prn:	noun: book cat: sn fnc: mdr: prn:	verb: . cat: v' decl prn:

syntactic−semantic parsing

1	noun: John cat: snp fnc: mdr: prn: 32	verb: try cat: n' a' v arg: mdr: prn:	*cross−copying*		
2	noun: John cat: snp fnc: try mdr: prn: 32	verb: try cat: #n' a' v arg: John mdr: prn: 32	verb: v_1 cat: inf fnc: arg: prn:	*cross−copying*	
3	noun: John cat: snp fnc: try mdr: prn: 32	verb: try cat: #n' #a' v arg: John v_1 mdr: prn: 32	verb: v_1 cat: ~~inf~~ fnc: try arg: John prn: 32	~~verb: read~~ ~~cat: n−s3' a' v~~ arg: mdr: prn:	*absorption with simultaneous substitution*

4	noun: John cat: snp fnc: try mdr: prn: 32	verb: try cat: #n' #a' v arg: John read mdr: prn: 32	verb: read cat: a' inf fnc: try arg: John prn: 32	noun: n_1 cat: sn' snp fnc: mdr: prn:	*cross−copying*

5	noun: John cat: snp fnc: try mdr: prn: 32	verb: try cat: #n' #a' v arg: John read mdr: prn: 32	verb: read cat: #a' inf fnc: try arg: John n_1 prn: 32	noun: n_1 cat: sn' ~~snp~~ fnc: read mdr: prn: 32	~~noun:~~ book ~~cat: sn~~ fnc: mdr: prn:	*absorption with simultaneous substitution*

6	noun: John cat: snp fnc: try mdr: prn: 32	verb: try cat: #n' #a' v arg: John read mdr: prn: 32	verb: read ~~cat: #a' inf~~ fnc: try arg: John book prn: 32	noun: book ~~cat: snp~~ fnc: read mdr: prn: 32	verb: . ~~cat:~~ v' decl prn: *absorption*

result

noun: John cat: snp fnc: try mdr: prn: 32	verb: try cat: #n' #a' decl arg: John read mdr: prn: 32	verb: read cat: #a' inf fnc: try arg: John book prn: 32	noun: book cat: snp fnc: read mdr: prn: 32

The use of read as an infinitive is based on the lexical proplet representing the non-third person singular present tense form of the verb with the feature [cat: n-s3′ a′ v].[15] Because the surface of this form may be used systematically[16] as the infinitive, it would violate the principle of surface composition-

---

[11] We are referring here to an elementary verb form such as (to) see, leaving phrasal constructions such as (to) have seen or (to) have been seen aside.

[12] For an analysis of infinitive constructions in different languages, see Wurmbrandt (2001) and the review by Reis and Sternefeld (2004).

[13] The forms used for the infinitive vary widely between languages. For example, German uses the first and third person plural present tense form, e.g. lesen. Latin has a separate surface for the infinitive, e.g. legere, which is distinct from any finite form (true infinitive).

[14] In many grammatical functions, the infinitive may be viewed as a stripped-down version of corresponding subclauses, e.g. as a subject sentence or an object sentence (15.6.1). However, subclauses have a finite verb and a separate prn value. Infinitives, in contrast, borrow tense, mood, and subject agreement from their finite matrix verb and share their prn value with the matrix.

[15] The cat values match the RegEx pattern (n-s3′ .* v), which represents the unmarked form of the finite verb. The Kleene star .* generalizes over varying oblique valency positions. RegExs were used in the NEWCAT implementations of LA grammar for matching ordered category values (Sect. 12.3). The equivalent DBS notation is [cat: n-s3′ X v]. Based on a restricted variable, here X, it is a more differentiated method than RegEx.

[16] The single exception is the auxiliary be. Unlike walk, for example, be has no finite form in the present tense which could serve as the citation form. Instead there are different forms for the first and third singular present, namely am and is with the cat values ns1′ and ns3′, and a third form are with the cat value ns2p′. In the past tense, there are the forms was and were with the cat values ns13′ and ns2p′, respectively. In addition there is the separate citation form be, which is a true infinitive in English because, unlike walk, it has no alternative use as a finite verb.

This is in contradistinction to all the other verbs of English, including do and have. They use two forms, e.g. have and has, for the present tense, and a single form, e.g. had, for the past tense. The latter has the cat value n′ for all numbers and persons of the nominative valency position. In summary, the computational treatment of subject/predicate agreement in English is based on distinguishing nominative valency positions, represented by the first cat value of a finite verb. Implementing the ality (1.4.3) to postulate separate lexical analyses for (a) the unmarked finite verb form and (b) the infinitive, imperative, and subjunctive of English.[17]

The content resulting from the derivation 15.4.1 is a set of proplets:

15.4.2 Content representation of an infinitive construction

$$\begin{bmatrix}\text{noun: John}\\ \text{cat: snp}\\ \text{fnc: try}\\ \text{mdr:}\\ \text{prn: 32}\end{bmatrix} \begin{bmatrix}\text{verb: try}\\ \text{cat: \#n}'\ \text{\#a}'\ \text{decl}\\ \text{arg: John read}\\ \text{mdr:}\\ \text{prn: 32}\end{bmatrix} \begin{bmatrix}\text{verb: read}\\ \text{cat: \#n}'\ \text{\#a}'\ \text{inf}\\ \text{fnc: try}\\ \text{arg: John book}\\ \text{prn: 32}\end{bmatrix} \begin{bmatrix}\text{noun: book}\\ \text{cat: snp}\\ \text{fnc: read}\\ \text{mdr:}\\ \text{prn: 32}\end{bmatrix}$$

John has try as its fnc value, while *try* has John as its first arg value; using schema (1) in 7.3.3, the duplex relation between *John* and *try* is characterized as subject/predicate. The proplet *read* has book as its second arg value, while *book* has read as its fnc value; using schema (2) in 7.3.3, the duplex relation between *read* and *book* is characterized as object\predicate.

There remains the duplex relation between *try* and *read*: one direction is coded by the value read in the second position of the arg attribute of *try*. This is opaque (NLC 6.6.5–6.6.7) insofar as read is verbal rather than nominal, as shown by the core attribute of the third proplet. The other direction is coded by the fnc attribute of *read* with the value try. It is opaque insofar as a lexical verb proplet does not have a fnc attribute. In 15.4.1, the fnc attribute is provided by the proplet representing the function word *to*, which copies its core value v_1 into the second arg slot of *try* in line 2. In line 3, all occurrences of v_1 are replaced by the core value of the *read* proplet, which is then discarded.

Because the object\predicate relation between *to read* (infinitival object) and *try* (matrix verb) is opaque, it is not characterized by any of the schemata in 7.3.3. Let us therefore define the following schema for the opaque intrapropositional object\predicate relation V\V for infinitival objects:

15.4.3 Pattern characterizing an elementary V\V signature

(9) verb\verb

$$\underset{\textit{to read}}{\begin{bmatrix}\text{verb: } \beta\\ \text{fnc: } \alpha\\ \text{arg: } \gamma\ \text{X}\end{bmatrix}} \underset{\textit{try}}{\begin{bmatrix}\text{verb: } \alpha\\ \text{arg: } \gamma\ \beta\ \text{Y}\end{bmatrix}}$$

(examples of matching content proplets for illustration only)

The number (9) refers to the corresponding elementaray signature in 7.6.4. The infinitive (first pattern) serves as an object of the matrix verb (second pattern).

proper variable restrictions is just another instance of linguistic watchmaker work in DBS.

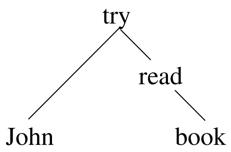
[17] Huddleston and Pullum (2002) postulate two lexical entries for the form in question, called "primary plain" and "secondary plain," and use the latter for the imperative, the subjunctive, and the infinitive.

More specifically, the variable β appears in the core attribute of the infinitive and in the object position of the arg slot of the matrix verb, while the variable α appears in the core attribute of the matrix verb and in the fnc slot of the infinitive. Furthermore, the variable γ representing the subject of the matrix verb appears in the arg slot of the infinitive as subject, coding the implicit subject of the infinitive in the result of 15.4.1 (subject control).

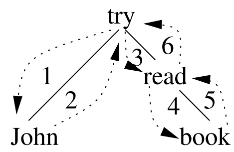
Based on the proplet set 15.4.2 and the schemata 1 and 2 of 7.3.3 and 9 of 15.4.3, the procedure described in Sect. 7.3 automatically derives the *(i) SRG*, the *(ii) signature*, and the *(iii) NAG* of the following DBS graph analysis:

15.4.4 DBS graph analysis of content derived in 15.4.1

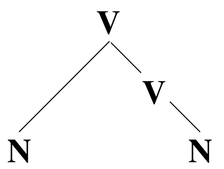
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	3	4	5	6
John	tried	to__read	a__book		.
V/N	N/V	V\V	V\N	N\V	V\V

The content 15.4.2 used for the language production shown by the *(iv) surface realization* resulted from the DBS.Hear derivation 15.4.1.

Using a content derived in the hear mode for a speak mode derivation back into the original surface is the ideal linguistic laboratory setup (14.4.1; NLC 3.5.2; HBTR 2.6.3) for modeling the communication cycle in a given language. For meaningful dialog, however, producing in the speak mode what has just been analyzed in the hear mode is not the most plausible scenario. Let us therefore consider language production from the following nonlanguage content:

> John and Mary are on a train in quiet conversation. At a station stop, the train car is suddenly stormed by a group of excited eleven-year-olds, making further talk impossible. Later Mary reports John's reaction as John tried to read a book.

How does Mary obtain the nonlanguage content for her utterance?

For the sake of the argument, let us replace Mary with an artificial agent. To obtain the nonlanguage content in question, the agent must apply its recognition to the raw data provided by the scene described. The purpose is to (i) establish the nodes *John, try, read*, and *book* as N, V, V, and N proplets, respectively, and (ii) to correctly establish the relations N/V, V\V, and N\V between them (8.1.1).

This process is impeded by the ambiguities caused by the {N, V} and {V, V} input pairs listed in 8.1.3. As an empirical means to improve efficiency by disambiguation, *selectional constellations* existing in the language between relevant classes of concepts may be used. How to obtain these selectional constellations from a corpus is the topic of the next section.

15.5 Selectional Constellations of Elementary Signatures

Different kinds of infinitive constructions depend on the higher verb (15.5.2, 15.6.3, 15.6.6, 15.6.8). For example, try and appear are both two-place verbs; but try may take an infinitive or a noun as object, while appear may take an infinitive only; try and appear are alike in that the implicit subject of the infinitive equals the higher subject (subject control).

Another minimal pair are promise and persuade. They are both three-place verbs, but promise may take an infinitive or a noun as an object, while persuade takes an infinitive only. Furthermore, the implicit subject of the infinitive equals the higher subject in promise (subject control), but the other higher object in persuade (object control).

The verb serving as the infinitival object, e.g. sleep in try to sleep, is unrestricted regarding its oblique valencies. This may be illustrated as follows:

15.5.1 Variation of oblique fillers in an infinitive

1. nominal object: John tried a cookie.
2. one-place infinitive object: John tried to sleep.
3. two-place infinitive object: John tried to read a book.
4. three-place infinitive object: John tried to give Mary a kiss.
5. infinitive with prepositional object: Julia tried to put the flower in a vase.
6. infinitive with object sentence recursion: Julia tried to say that Bill believes that Mary suspects that Susy knows that Lucy loves Tom.
7. ...

As shown by examples 2–6, the verb representing the infinitive may be one-place, two-place, or three-place; take a prepositional object, an object sentence, an iteration of object sentences; etc.

In DBS, verbs which have the same valency structure constitute a lexical class. Formally, such a class may be characterized by one or more patterns which are matched by each of its elements. For naming the class, we choose the surface of a representative element, for example try. Consider the two patterns defining the try class:

15.5.2 BNC-BASED DEFINITION OF try CLASS INFINITIVES

noun\verb	verb\verb	
$\begin{bmatrix}\text{noun: } \beta \\ \text{fnc: } \alpha\end{bmatrix} \begin{bmatrix}\text{verb: } \alpha \\ \text{arg: } \gamma\ \beta\end{bmatrix}$	$\begin{bmatrix}\text{verb: } \beta \\ \text{fnc: } \alpha \\ \text{arg: } \gamma\ \text{X}\end{bmatrix} \begin{bmatrix}\text{verb: } \alpha \\ \text{arg: } \gamma\ \beta\end{bmatrix}$	
cookie *try*	*to read* *try*	(examples of matching content proplets for illustration only)

where $\alpha \in$ {*begin, can afford, choose, decide, expect, forget, learn, like, manage, need, offer, plan, prepare, refuse, start, try, want*}[18]

Selectional constellations:

matrix verb α	*subject* γ	*nominal object* β
begin 6 642	government 32, people 26, commission 24, ...	work 206, career 141, life 113, ...
can afford 1 542...		...
...		

matrix verb α	*subject* γ	*infinitival object* β
begin 18 992	people 204, men 55, government 54, number 49, ...	feel 528, be 492, take 371, ...
can afford 1 841 ...		...
...		

The numbers specify the associated frequencies found in the BNC. The N/V and V/V patterns in combination with the variable restrictions may be seen as a revival of Fillmore's (1968) case frame idea, though in a different form.

The elements in a lexical class are named by elementary signatures. For example, the schema on the left in 15.5.2 is named noun\verb; it consists of the higher verb α which takes the noun γ as subject and the noun β as object. In the schema on the right, named verb\verb, the higher verb α takes the noun γ as subject and the verb β as object; the infinitive β takes the higher verb α as its fnc value and the subject γ of α as its first arg value (subject control).

The verb of an infinitive as subject (V/V, 7.6.4 (8)) or as object (V\V, 7.6.4 (9)) may have a free range of objects, as illustrated in 15.5.1, 2–6. Similar to the schema 15.4.3, this range is indicated by the variable X in the feature ([arg: γ X]).

The two schemata for the try class are sufficiently distinctive for querying a corpus database like 15.5.3 in order to establish a restriction list for the variable α automatically. The resulting list will consist of all the verbs in the corpus, here the BNC, which have the same grammatical properties as try.

Similar restrictions may be established for each individual element in the α list regarding its subject variable γ and its object variable β, for objects as N and as V. For example, as indicated in 15.5.2, the higher verb begin occurs

[18] The restrictions on the variable α were gleaned from Sorensen (1997). In contradistinction to the *selectional restrictions* of generative grammar (e.g. Klima 1964), the *selectional constellations* of DBS record the distributional facts in a corpus. Thanks to T. Proisl for determining the BNC frequencies in 15.5.2 (*try*), 15.5.3 (*decide*), and (15.6.2 (bare infinitive).

in such selectional N\V constellations as government begins work, people begin career, etc.; corresponding V\V constellations are people begin to feel, men begin to be, and government begins to take.

In summary, definition 15.5.2 refines 15.4.3 by (i) listing all possible higher verb concepts of the try class as a restriction on the variable α. It (ii) provides schemata for an object as (i) a noun or (ii) a verb used as an infinitive. And it (iii) lists the selectional constellations between each higher verb α, its subjects γ and its nominal or infinitival objects β as n-tuples, for n=3. The numbers represent the BNC frequencies for each constellation, in decreasing order.

The basis for computing variable restrictions and frequencies in a lexical class entry like 15.5.2 is a DBS *corpus word bank*. For simplicity, the following example is limited to the try class verb decide, showing the token lines for its grammatical noun objects with BNC frequencies:

15.5.3 Nominal decide objects in a DBS corpus word bank

member proplets	*owner values*
. . . [noun: case fnc: decide frq: 65] . . .	case
. . .	. . .
. . . [verb: decide arg: X game frq: 4] [verb: decide arg: X dispute frq: 7] [verb: decide arg: X fate frq: 47] [verb: decide arg: X case frq: 65] [verb: decide arg: X issue frq: 74]	decide
. . .	. . .
. . . [noun: dispute fnc: decide frq: 7] . . .	dispute
. . .	. . .
[noun: fate fnc: decide frq: 47]	fate
[noun: game fnc: decide frq: 4] . . .	game
. . .	. . .
. . . [noun: issue fnc: decide frq: 74]	issue

The member proplets in a token line are ordered according to increasing frq (frequency) values from left to right (and not by the order of arrival).[19] The example shows that decide occurs 65 times with the word case as object and that case occurred 65 time with the functor decide in the BNC.

In contradistinction to an episodic word bank (4.1.1), a DBS corpus word bank like 15.5.3 is static – which is why it doesn't have a now front (compare 15.5.3 with 4.1.1, 6.4.3, Sects. 13.3–13.5). The content of a DBS corpus word bank is changed only after more corpus work has revealed additional argument values for a given verb, additional functor values for a given noun, and so on, as well as new frequency numbers, such that the elements and their order in a token line have to be adjusted (recomputed).

Querying a DBS corpus word bank is based on pattern proplets. The query pattern is matched with elements of a corresponding token line from right to left, i.e. from more frequent to less frequent, until a successful match is found (Sect. 4.2). For example, the question *How often does* decide *occur with the nominal object* case *in the BNC corpus?* may be formalized as the following query:

15.5.4 Formal query and answer 1

$$\begin{array}{ll} \textit{query} & \textit{result} \\ \begin{bmatrix}\text{verb: decide}\\ \text{arg: X case}\\ \text{frq: ?}\end{bmatrix} & \begin{bmatrix}\text{verb: decide}\\ \text{arg: X case}\\ \text{frq: 65}\end{bmatrix} \end{array}$$

In the result, the “?” of the query is replaced by the frq value searched for.

Similarly, the question *How often does* case *occur as a nominal argument of* decide *in the BNC corpus?* may be formalized as the following query:

15.5.5 Formal query and answer 2

$$\begin{array}{ll} \textit{query} & \textit{result} \\ \begin{bmatrix}\text{noun: case}\\ \text{fnc: decide}\\ \text{frq: ?}\end{bmatrix} & \begin{bmatrix}\text{noun: case}\\ \text{fnc: decide}\\ \text{frq: 65}\end{bmatrix} \end{array}$$

If case occurred also as the subject of decide, as in This case decided the issue, the query would return two answer proplets, one for case as subject, the other for case as object, each with its respective frq value.

Finally consider *Which intrapropositional relations in the BNC use* decide*?*

15.5.6 Formal query and answer 3

$$\begin{array}{llllll} & & & & \textit{result}^{20} & \textit{query} \\ \begin{bmatrix}\text{noun: game}\\ \text{fnc: decide}\\ \text{frq: 4}\end{bmatrix} & \begin{bmatrix}\text{noun: dispute}\\ \text{fnc: decide}\\ \text{frq: 7}\end{bmatrix} & \begin{bmatrix}\text{noun: fate}\\ \text{fnc: decide}\\ \text{frq: 57}\end{bmatrix} & \begin{bmatrix}\text{noun: case}\\ \text{fnc: decide}\\ \text{frq: 65}\end{bmatrix} & \begin{bmatrix}\text{noun: issue}\\ \text{fnc: decide}\\ \text{frq: 74}\end{bmatrix} & \begin{bmatrix}\text{noun: ?}\\ \text{fnc: decide}\\ \text{frq:?}\end{bmatrix} \end{array}$$

[19] Including the prn feature of the original corpus will help to answer further linguistic questions.

[20] Normally, the query precedes the result, as in 15.5.4 and 15.5.5. When the result is a token line, however, we keep the order based on increasing frequency (15.5.3), followed by the query pattern.

Here the query contains two question marks. The result corresponds to the token line of decide.[21] The query answer begins with the most frequent item.

A schema like 15.5.2 and the corpus word bank on which it is based supports the agent's current speech production by providing empirically based frequencies for collocations (NLC Sect. 15.6). It also provides a search space reduction for recognition. For example, as soon as an item in the raw data has been recognized by a certain concept, e.g. *decide*, the matching procedure for the following raw surfaces first tries those concepts which are related to *decide* in the agent's word bank, beginning with the most frequent ones (Sect. 8.6; L&I'05). Like intersection in coactivation (Sect. 5.4), this process converges quickly.

Thus, corpus linguistics may complement the work on an artificial agent even before robotics makes the external interfaces available (Liu 2009a). Using concept names as placeholders (6.6.8), the selectional constellations illustrated in 15.5.2 and the frequency values may be obtained from any given corpus.

The method requires word form recognition for (i) *lemmatization*, i.e. reduction to the base (citation) form, and (ii) for *categorization*, i.e. determining the morphosyntactic properties. It also requires syntactic-semantic parsing for establishing the semantic relations of structure as listed in 7.6.4, 7.6.5.

15.6 Appear, Promise, and Persuade Class Infinitives

Consider the following content structures corresponding to an infinitive construction in English:

15.6.1 Content structures corresponding to infinitives

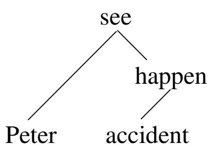
1. Infinitive as subject: To err is human, to forgive divine.
2. Infinitive as object: John tried to read a book.
3. Infinitive as adnominal modifier: the desire to help
4. Bare infinitive:[22] Peter saw the accident happen.

(1) is largely restricted to copula constructions (NLC 6.6.9–6.6.11), but there are also examples with a full higher verb, such as To plan for a vacation can be a drag. (2) has been illustrated in Sect. 15.4. (3) is a post-nominal variant of the V|N relation (7.6.4, 10). (4) has the following DBS graph analysis:

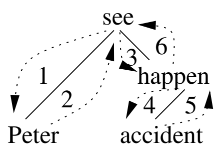
[21] Selectional constellations for given natural languages are of special interest in machine translation (Sect. 16.3; cf. Kay 1984, Dorr 1993, Wu and Palmer 1994, and many others). Storing selectional constellations in a corpus word bank may support this application with the efficient retrieval of highly differentiated collocation information based on pattern proplets connected by semantic relations.

15.6.2 BARE INFINITIVE: Peter saw the accident happen.

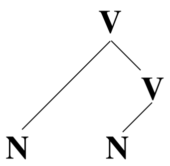
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	3	4	5	6
Peter	saw		the__accident	happen	.
V/N	N/V	V\V	V/N	N/V	V\V

At the surface, accident is a direct object[23] of the higher verb *see*, yet at the level of content the real object is the verb *happen* which takes accident as its subject (see SRG and signature). As far as the surface order is concerned, the post-nominative rule for the position of the verb in English declaratives (FoCL Chap. 17), i.e. accident happen(ed), coincides with the other role of accident as the post-verbal object of the higher clause, i.e. saw accident.

The resulting discrepancy between the grammatical appearance and the content construction resembles that of gapping (Sects. 9.5, 9.6): both constructions occur in many natural languages[24] with strong native speaker intuitions, but are not frequent in spoken everyday language.[25]

Each construction in 15.6.1 defines at least one class of infinitive-taking verbs. For example, in addition to the try class discussed in Sects. 15.4 and 15.5, there is also the appear class. It resembles the try class (15.5.2) in that it consists of proplets corresponding to transitive higher verbs which take an infinitive as object. It differs, however, in that the verbs in the appear class do not allow nouns as objects.

15.6.3 Appear CLASS INFINITIVE-AS-OBJECT CONSTRUCTIONS

1. nominal object: *John appeared a cookie.[26]

[22] Bare infinitives are so called because they lack the to. Modal constructions, as in Peter can see the horizon., do not reflect a separate kind of infinitive content construction and are treated like auxiliary constructions in DBS (NLC 6.6.8; FoCL Sect. 17.3),

[23] In the German translation Maria sah den Unfall geschehen, the counterpart of accident, i.e. den Unfall, is morphologically marked for accusative.

[24] The English bare infinitive corresponds to the Latin AcI, though with fewer selectional constellations.

[25] In the BNC, 0.2% of the sentences contain a bare infinitive.

[26] Exceptions are illustrated by examples like John appeared five times (frequency) and John appeared on the scene (locality). These nominal and prepositional objects are adverbial in charac-

2. one-place infinitive object: John appeared to sleep.
   . . . (as in 15.5.1)

These properties may be formalized as the following schema:

15.6.4 Definition of appear class infinitives

verb\verb

$$\begin{bmatrix}\text{verb: }\beta\\ \text{fnc: }\alpha\\ \text{arg: }\gamma\ \text{X}\end{bmatrix}\begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\gamma\ \beta\end{bmatrix}$$

to sleep *appear* (examples of matching proplets for illustration only)

where $\alpha\ \epsilon$ {agree, appear, be able, seem, tend}

Selectional constellations:[omitted]

The restrictions on the variable α for the appear and the try (15.5.2) class constructions are disjunct.

A third class of verbs which take infinitives as object is called the promise class. It differs from the try and appear classes in that verbs like promise have three valency positions, one for the subject, one for the indirect object, and one for the direct object. The latter may be either a noun or an infinitive.

15.6.5 Promise class infinitive-as-object constructions

1. nominal object: Mary promised John a cookie.
2. one-place infinitive object: Mary promised John to sleep.
   . . . (as in 15.5.1)

Accordingly, the promise class is defined with the following two schemata:

15.6.6 Definition of promise class infinitives

$$\begin{bmatrix}\text{noun: }\delta\\ \text{fnc: }\alpha\end{bmatrix}\begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\alpha\end{bmatrix}\begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\gamma\ \delta\ \beta\end{bmatrix}$$

John *cookie* *promise*

$$\begin{bmatrix}\text{noun: }\delta\\ \text{fnc: }\alpha\end{bmatrix}\begin{bmatrix}\text{verb: }\beta\\ \text{fnc: }\alpha\\ \text{arg: }\gamma\ \text{X}\end{bmatrix}\begin{bmatrix}\text{verb: }\alpha\\ \text{arg: }\gamma\ \delta\ \beta\end{bmatrix}$$

John *to sleep* *promise*

(examples of matching proplets for illustration only)

where $\alpha\ \epsilon$ {offer, promise, threaten[27]}

---

ter and illustrate once more the importance of recording and utilizing selectional constellations for each content word, based on a large RMD corpus (Reference Monitor corpus with Domain structure, Sect. 15.3).

[27] Threaten is an exception insofar as it doesn't take a noun in place of the infinitive as its second object, as in unacceptable *Mary threatened John a cookie. This may be handled as part of the selectional constellations, resulting in a separate class instead of an exception.

Selectional constellations:[omitted]

The schema on the left matches the three-place verb constructions with the elementary signatures N\V (indirect object) and N\V (direct object), while the schema on the right matches the elementary signatures N\V (indirect object) and V\V (infinitive).[28]

The core value of the indirect object, here John, is represented by δ. It appears in the second arg slot of the higher verb α. The first arg value, γ, of this attribute represents the subject of the higher verb and reappears in the subject slot of the infinitive verb β, thus coding subject control.

A fourth class of infinitive objects is called the persuade class. It differs from the promise class in that the higher verb may not take a noun instead of an infinitive. Also, the implicit subject of the infinitive is the nominal object of the higher construction (object control). Consider the following examples:

15.6.7 Persuade CLASS INFINITIVE-AS-OBJECT CONSTRUCTIONS

1. nominal object: *Mary persuaded John a cookie.
2. one-placе infinitive object: Mary persuaded John to sleep.
   ... (as in 15.5.1)

Object control means that it is the object John who sleeps, in contrast to the examples in 15.5.1, in which it is the subject.

While John is the indirect object in 15.6.5, it is the direct object in 15.6.7. This leaves the second arg position in the higher verb to the infinitive.[29]

15.6.8 DEFINITION OF persuade CLASS INFINITIVES

$$\begin{bmatrix}\text{verb: } \beta \\ \text{fnc: } \alpha \\ \text{arg: } \delta\ \mathrm{X}\end{bmatrix} \begin{bmatrix}\text{noun: } \delta \\ \text{fnc: } \alpha\end{bmatrix} \begin{bmatrix}\text{verb: } \alpha \\ \text{arg: } \gamma\ \beta\ \delta\end{bmatrix}$$

to sleep *John* *persuade* (examples of matching contents, for illustration only)

where α ϵ {advise, allow, appoint, ask, beg, choose, convince, encourage, expect, forbid, force, invite, need, permit, persuade, select, teach, tell, urge, want, would like}.

Selectional constellations:[omitted]

The grammatical role of the variable δ as the direct object is coded by its final position in the arg slot of the verb α. The restriction on the variable α in the

---

[28] The distinction between indirect and direct object is coded by the order of the arg values.

[29] Given that persuade class constructions do not allow replacing the infinitive with a nominal object, it does not necessarily follow that the infinitive in a persuade class construction must function as the indirect object of the higher verb. In German, for example, there are three-place verbs like lehren (teach) which take two objects in the accusative, as in Maria lehrte den Freund den Tango. This would correspond to two direct objects in English.

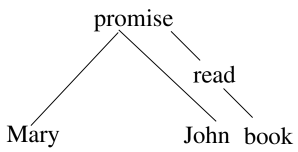
persuade class has substantially more elements than that in the promise class (15.6.6).[30]

The single schema matches two elementary signatures, N\V (noun as direct object) and V\V (infinitive as indirect object). The noun pattern matches the direct object, e.g. John. Its core value, represented by δ, appears in the third arg slot of the higher verb α and in the first arg slot of the infinitive β, thus coding object control.

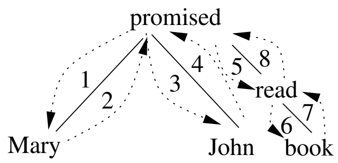
In conclusion let us compare the DBS graph analysis of a promise and a persuade class infinitive content construction:

15.6.9 Subject control in Mary promised John to read a book.

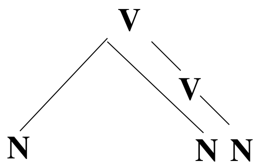
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	3	4	5	6	7	8
Mary	promised	John		to__read	a__book		•
V/N	N/V	V\N	N\V	V\V	N\V	V\N	V\V

The corresponding proplet representation specifies the implicit subject of the infinitive explicitly as Mary in the arg slot of *read* (subject control):

15.6.10 Corresponding proplet representation

noun: Mary	verb: promise	noun: John	verb: read	noun: book
cat: snp	cat: #n′ #d′ #v′ decl	cat: snp	cat: #n′ inf	cat: snp
sem: nm f sg	sem: past	sem: nm f sg	fnc: promise	sem: sg
fnc: promise	arg: Mary John read	fnc: promise	arg: Mary book	fnc: promise
prn: 35	prn: 35	prn: 35	prn: 35	prn: 35

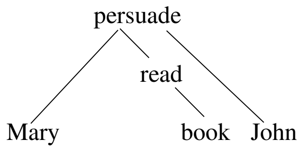
Similar to the graphs of the promise class are those of the persuade class constructions except that the order of the infinitive and the nominal object is reversed, indicating object control.[31] Consider the following example:

[30] Comrie's (1986) observation that subject and object control are determined also by pragmatic factors may be accommodated by nondisjunct restriction sets on the variable α in various class definitions.

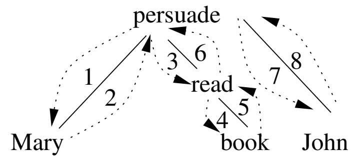
[30] In German, the distinction between the direct and the indirect object may be expressed by accusative vs. dative case markings of the respective object. For example, Maria versprach dem (dative, 15.6.9) Freund zu singen., but Maria überredete den (accusative, 15.6.11) Freund zu singen

15.6.11 Object control in Mary persuaded John to read a book.

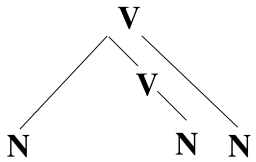
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	7	8	3	4	5	6
Mary	persuaded	John		to__read	a__book		•
V/N	N/V	V\N	N\V	V\V	N\V	V\N	V\V

The implicit subject of the infinitive is the direct object John (object control). The corresponding proplet representation shows John in the arg slot of *read*:

15.6.12 Corresponding proplet representation

noun: Mary	verb: persuade	noun: John	verb: read	noun: book
cat: snp	cat: #n′ #inf′ #a′ decl	cat: snp	cat: #n′ #a′ inf	cat: snp
sem: nm m sg	sem: past	sem: nm f sg	fnc: persuade	sem: indef sg
fnc: persuade	arg: Mary read John	fnc: read	arg: John book	fnc: read
prn: 36	prn: 36	prn: 36	prn: 36	prn: 36

The described method of determining the selectional constellations in a corpus depends crucially on the binary treatment of semantic relations in DBS. The method can not be used by systems computing possible substitutions such as PS grammar and C grammar. This is because a substitution-based derivation of phrases and clauses is inherently compelled to use nonterminal nodes, resulting in non-binary (indirect) relations between the terminal nodes.

In a big corpus, the selectional constellations represent the content word distributions of many domains and many authors all mixed up. This may be instructive for what might be regarded as the language "as a whole" at a certain time. It may be equally instructive, however, to analyze the word form distributions in a certain domain or in the work of a single author.[32]

Determining the selectional constellations for a corpus, a domain, or an author is based on the same routine in DBS. It consists of automatic word form recognition, syntactic-semantic parsing, frequency analysis, and storage in a DBS corpus word bank.

[32] Word distributions have in fact been used to argue for or against authorship in controversial cases such as Shakespeare.

Remark on Different Kinds of Corpora

Corpora containing *practical* language data are exemplified by the correspondence and documentation of a company (Blair and Maron 1985) or the content of the Internet. Their purpose is to answer genuine user queries with optimal recall and precision (FoCL 2.1.3). Though not designed originally for interacting with textual databases, DBS is well-suited for the purpose: the hear mode may be used for parsing text into content and for interpreting natural language queries; the inferencing of the think mode may be used for expanding the contents of query answers; and the speak mode may be used for query answering in the natural language of choice. Compared to building a talking robot, the application is simplified because it gets by with place holder (instead of procedurally implemented) core values (6.6.8).

Corpora containing *theoretical* language data, in contrast, are exemplified by the British National Corpus (BNC, Burnard 1995) and the Corpus of Contemporary American English (COCA, Davies 2008). Their content is of little interest to the linguists who build and study them. Instead, the focus is on analyzing different constructions, collocations, idioms, and the frequency distribution of word forms in the natural language at hand.[33] Answering content-oriented user queries is not an intended application.[34]

Practical and theoretical corpora have in common that they consist of written language. The DBS methods developed for processing spoken language, as in spontaneous dialog (Chap. 10), may be re-used for the querying and query answering of written language in practical and theoretical (15.5.4–15.5.6) corpora. Conversely, the methods developed for the analysis of written language stored in theoretical corpora, such as statistical tagging, in contrast, can not be extended to a real time on the fly parsing of life language in spoken dialog.

[33] Corpus linguists take pride in analyzing "real" data, such as newspaper articles, in contrast to examples constructed by language scholars for illustrating a grammatical property as clearly as possible. The insistence on real data may have been in reaction to some bizarre claims within nativism, for which invented borderline examples prefixed with *In my dialect* ... were offered as "proof."

We should note, however, that corpus data and reasonable invented language examples are instances of the same linguistic intuitions within the language community. Corpus data (also in the form of systematic test lists) are most suitable for the verification of a current DBS software version, while invented examples are most suitable for the systematic upscaling to the next version.

[34] The corpora containing theoretical language data may be further differentiated into one-shot corpora consisting of data collected in a given period of time, such as the Brown Corpus and the BNC, and complex corpora consisting of a reference corpus followed by a sequence of monitor corpora, such as the COCA. Also, there is the distinction between corpora which aim at providing an accurate snapshot of a natural language trying to be representative and balanced (FoCL 15.3.1), and "opportunistic" corpora, which leave it to the user to pick from the data pile what seems expedient.

16. Practical Benefits of Basic Research

The DBS software machine may be used (i) with procedurally defined core values or (ii) with place holders. Procedural core values require a robot with a real body and external interfaces for recognition and action, capable of constant interaction with a real environment (grounding). This is a precondition for immediate reference (FoCL 4.3.2; NLC 2.5.1; HBTR 3.2.5)

The use of DBS with place holder core values, in contrast, may be implemented on a standard computer, using the keyboard and the screen as the preliminary interfaces for recognition and action. As a consequence, only mediated reference (FoCL 5.3.4; NLC 2.4.1; HBTR 3.2.4) is possible. In either case, the artificial interfaces have to be compatible with those of the human prototype (2.6.2).

Place holder core values may be used for today's natural language processing (NLP) such as retrieval from online text data, querying, query answering, corpus linguistics, and machine translation. For these applications, the machine is not required to understand the language being processed, though it would certainly help.

16.1 Extending the Data Coverage

The DBS approach to practical (commercial) applications of natural language processing is based on solving the most important theoretical question first:

16.1.1 THE SIXTY-FOUR-THOUSAND-DOLLAR QUESTION OF NLP

How does the mechanism of natural language communication work?

To protect against accidentally neglecting some crucial interface, component, or ability, the overall design of a DBS robot aims at functional completeness. By modeling all essential structural aspects of human natural language communication it is hoped that there will be no application-motivated requests which cannot be accommodated because of incomplete linguistic functionality.

If such a software system works well at the proper levels of abstraction, though initially with small (and highly relevant) data coverage only, then all that remains to be done is to increase the data coverage. For natural language communication, this may seem like a mammoth project, though nothing compared to projects in physics (CERN) or biology (human genome project), for example. Nevertheless, work in computational linguistics should be similar to these projects in that solid solutions (FoCL Sect. 2.3) are combined into a coherent foundation which supports near completeness of function and data coverage for different natural language and domains over the long term.

Thereby pure research will have immediate consequences for the natural language processing needs of commercial users. Take for example LA morph, the automatic word form recognition software (FoCL Chaps. 13–15), running in real time with a certain natural language of choice (3.5.2). Its data coverage may be extended by adding to the lexicon and by optimizing the allo- and combi-rules (Sect. 2.5). In addition to broadening the base for syntactic-semantic analysis and for inferencing, it provides applications with better retrieval results: different word forms, e.g. swam and swimming, are recognized as the same word, i.e. *swim*, based on lemmatization (FoCL 13.4.3).

A second area for improving the data coverage is extending the syntactic-semantic analysis. When applied to a new (i.e. previously unanalyzed) natural language, the DBS.Hear parser will at first handle only a few constructions (NLC Chaps. 11, 13). As the language is being studied, more and more constructions are added to the grammar, tested, and improved. When the DBS.Hear parser encounters input it cannot yet handle, the passage may be traversed at a lower level of detail until proper parsing may resume (robustness). For this, LA grammar is especially suitable because it computes possible continuations in a time-linear derivation order.

Expanding syntactic-semantic parsing in the agent's hear mode is more demanding than automatic word form recognition. This effort shall not go unrewarded from the application side, however. The coding of functor-argument and coordination relations extends retrieval from word forms to phrases and clauses, and from there to sentences, paragraphs, and text. Technically, this amounts to an extension from matching analyzed content words stored within token lines to matching semantic relations defined between content words across token lines in the word bank.

A third area for extending the data coverage is the think mode, which combines (i) selective activation by DBS.Nav and (ii) inferencing by DBS.Inf (Chap. 5). Selective activation is implemented as a navigation along the semantic relations in a word bank[1] (coactivation, Sect. 5.4), serving to activate and report stored content. Inferences are used for deriving the different perspectives of the speaker and the hearer on content, and to compute blueprints for action, including language action. Together with current and stored data, coactivation (13.4.4) and inferencing (13.5.3) constitute the agent's autonomous control, which has many practical applications, with and without language.

Finally, consider DBS.Speak. Defined as a language-dependent variant of DBS.Nav and DBS.Inf, DBS.Speak embeds language-dependent lexicalization rules into the sur slots of the patterns.[2] Utilizing the navigation provided by DBS.Nav and DBS.Inf as the macro word order (Chap. 7), the surface production of DBS.Speak takes care of the micro order and the proper perspective (e.g. tense) and morphosyntactic adjustments (e.g. agreement).

This work will not go unrewarded from the application side either. The obvious application is query answering in natural language. Thereby the DBS.Speak part is only the tip of the iceberg. Prior to answering, the query is converted automatically into schemata which are used to coactivate corresponding contents. These are processed into answer contents by means of inferencing (Sect. 14.3) and realized by the lexicalization rules of DBS.Speak as unanalyzed external surfaces in the modality and the language of choice.

16.2 Getting Applications to Fund Foundations

While specific applications may benefit selectively from nurturing a particular component, all applications will benefit simultaneously from a systematic upscaling of the DBS robot as a functionally coherent whole. If an application does not require certain abilities, they may be switched off.[3]

The systematic transfer from a continuously improving DBS system to commercial applications of natural language processing and human-machine communication may be illustrated by the following vision: Every year, when the current monitor corpus (Sect. 15.3) has been put through the software grinder of automatic word form recognition, syntactic-semantic parsing, frequency

[1] A word bank may be viewed as a syntactic-semantic network, though restricted to the classic relations of functor-argument and coordination, at the elementary, the phrasal, and the clausal levels of grammatical complexity (Sect. 9.6). For some questions and results of linguistic networks, see Liu (2011), Solé et al. (2010), Sowa (1987/1992), Brachman (1979), and others.

[2] Chap. 10 shows the interaction between DBS.Nav, DBS.Inf, and DBS.Speak in a fictional dialog between Jean Paul Sartre and Simone de Beauvoir.

[3] For example, a dialog system over the phone may omit the ability to read.

analysis, and comparison with preceding monitor corpora, the results are used for producing a software version with corrected and extended data coverage.

By regularly making new versions available to paying subscribers for their natural language processing needs, all or most of the research costs may be recovered. For this to work long-term, a new release must not require any labor (e.g. additional personnel training) from the subscriber.[4] Also, each new version must enhance service directly and noticeably, so that paying subscribers are attracted and kept in sufficient numbers.

Improvements from one version to the next may be achieved rather easily because there are large fields of empirical data which merely need to be "harvested." The harvester is the DBS software machine. Originally designed to model the natural language communication mechanism in the form of a talking robot, its off-the-shelf components for the lexicon, word form recognition, syntactic-semantic parsing, and so on, may be used for storing the language-dependent data of a new language: words are added to the robot's lexicon component, just as compositional structures are added to LA Morph, DBS.Hear, DBS.Nav, DBS.Inf and DBS.Speak in the robot's rule component. Also, culture-, domain-, and application-dependent content may be added to the word bank and new inferences may be defined.

Storing the new or revised language analyses directly in the DBS robot makes the harvest immediately available (i) for computational testing by pure research and (ii) for computational applications by the users. This holds not only for the hear mode, as in testing on theoretical and practical data (Chap. 15, concluding remark), but for the full cycle of language communication.

There is no lack of renewable language data for long-term linguistic research, namely (i) the natural changes from year to year within the domains of a given language, based on the RMD corpus, and (ii) a wide, constantly extending range of applications in human-machine communication and natural language processing. In addition, there is (iii) the great number of natural languages not yet charted, or not yet charted completely, including English.

Charting a new natural language in DBS is a standard procedure, but it has to deal with relatively large amounts of data. As more and more languages are analyzed, however, charting is accelerated because software constructs may be reused, based on similarities in lexicalization, in productive syntactic-semantic structures, in collocations, constructions, and idioms, and in inferencing. To better support day-to-day research,[5] these standardized software constructs

[4] Most comfortable for the user would be providing updates automatically by means of patch computing.

[5] For example, in work on language typology or upscaling to additional data in a given language.

and their declarative specifications may be stored in system libraries, organized for families of languages.

16.3 Machine Translation

Among the applications of DBS there are those which use the cycle of natural language communication almost completely, for example, the storage and query of practical language data. Then there are applications which use only part of DBS, while adding theoretical requirements of their own, for example, corpus linguistics. And there is an application which uses the full communication cycle not only for one language but for several while requiring substantial additional skills, namely machine translation (Lenders 2012).

DBS approaches machine translation by analyzing multilingual communication as characteristic sequences which are built from those of monolingual communication. Characteristic sequences have been analyzed in Chap. 10 (esp. 10.6.6, 10.6.7) for statement, Yes-No question, WH question, and request. In preparation for analyzing translation, monolingual dialog may be shown abstractly as the following characteristic sequences.

16.3.1 ELEMENTARY MONOLINGUAL COMMUNICATION EVENT

Speak mode event: A speak C in L
Hear mode event: B hear C in L

where A and B are agents, C is a content, L is a language, and speak and hear are used modality-free.

Characteristic sequences start with the speaker's selection or derivation of content and end with the hearer's storage and interpretation. As an act of communication, a characteristic sequence is successful if the content coded by the speaker is reconstructed (i) equivalently and (ii) with the appropriate STAR coordinates by the hearer (Chap. 10).

The simplified sequence 16.3.1 may be extended to monolingual face-to-face dialog. Let C and C' be different contents:

16.3.2 ELEMENTARY TURN-TAKING (MONOLINGUAL)

A speak C in L, B hear C in L;
B speak C' in L, A hear C' in L;
repetitur

Monolingual reading of a text, in contrast, consists of the following characteristic sequences:

16.3.3 Monolingual reading a text aloud

A speak C in L, B hear C in L;
A speak C' in L, B hear C' in L;
repetitur

In 16.3.2 and 16.3.3, each characteristic sequence consists of a pair of lines, which may be repeated with new C and C'. Each line consists of two basic communication units, namely a speak mode event followed by a hear mode event. The difference between the characteristic sequences 16.3.2 and 16.3.3 is the distribution of the agents A and B in the respective second line.

There is also a variant in which a multilingual agent A says something in L (e.g. English) and an equally multilingual agent B replies in L' (e.g. French). This bilingual case may be represented abstractly as follows:

16.3.4 Bilingual face-to-face dialog with turn taking

A speak C in L, B hear C in L;
B speak C' in L', A hear C' in L';
repetitur

Bilingual machine translation and its natural counterparts differ from the monolingual variants 16.3.2 and 16.3.3 and the multilingual variant 16.3.4 in that their characteristic sequences consist of four lines rather than two. More specifically, if A, B, and T (for translator) are agents, C and C' are contents, and L and L' are languages, then the characteristic sequences of simultaneous interpretation (for spoken[6] dialog) and translation (for reading) are as follows:

16.3.5 Bilingual simultaneous interpretation and translation

1. *Simultaneous interpretation*

   A speak C in L, T hear C in L;
   T speak C in L', B hear C in L';
   B speak C' in L', T hear C' in L';
   T speak C' in L, A hear C' in L;
   repetitur

2. *Translation*

   A speak C in L, T hear C in L;
   T speak C in L', B hear C in L';
   A speak C' in L, T hear C' in L;
   T speak C' in L', B hear C' in L';
   repetitur

The difference between simultaneous interpretation and translation is in the respective lines 3 and 4. Apart from the role of the translator, simultaneous interpretation resembles 16.3.2, while translation resembles 16.3.3.

16.4 Achieving Equivalence between Source and Target

The pivot of simultaneous interpretation and of translation is agent T, i.e. the interpreter-translator. The communicative goal of T is the *equivalence* between the source and the target language representation of the content provided by the original speaker. This is a specialization of the general goal of agents, i.e. maintaining a state of balance. In contrast to normal communication, T does not react to the original speaker's content directly. Instead T's balance is driven by a desire to do a professional job and get recompensed for it.

As an example of simultaneous interpretation, consider agent A who has just escaped from a plane crash. A is from a foreign country and describes the incident in his or her native language L to an interpreter T for investigator B. The words representing the experience, i.e. content C, gush out of A automatically, describing the events in their temporal order and/or according to A's individual assessment of importance. This process is enabled by the close correlation between content coded at agent's A context and language levels.[7]

Agent T, who speaks A's native language L, interprets A's description in the language L′ of the land. While A uses one language, L, T has to use two, L and L′. The difficulty for the human translator as well as for machine translation is achieving content equivalence between the coding in L vs. L′. In DBS, achieving this equivalence is facilitated by the following structural operations which are integral parts of the cycle of natural language communication anyway:

16.4.1 Reducing language differences in content coding

In a language content,

1. the surface order is replaced by order-free[8] proplets,
2. function words and inflectional/agglutinative markings are eliminated by encoding their semantic contributions as grammatical values in content proplets, and

[6] We are leaving aside such variants as written dialog, as in an exchange of letters.

[7] Reference, Sect. 4.3; FoCL. Sect. 5.4; NLC 3.2.4; HBTR Chap. 3.

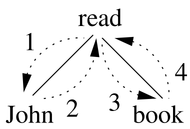
[8] The assumption of an order-free content seems to agree with the SemR level of the Meaning-Text theory (MT) proposed by Zholkovskij and Mel'chuk (1965). An order-free level is also used in some dependency grammars, e.g. Hajičová (2000).

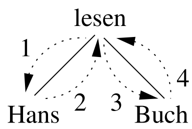
3. semantic relations between content proplets are encoded as elementary signatures (7.6.4–7.6.5) by means of proplet-internal addresses.

In the following example, the source and the target language have the same content representation (16.4.3), but differ in word order:

16.4.2 Word order difference

English word order *German word order*





1	2	3	4
John	has_read	the_book	.
V/N	N/V	V\N	N\V

1	2	3	4
Hans	hat	das_Buch	gelesen_.
V/N	N/V	V\N	N\V

In English, the complex verb form has_read is realized in navigation step 2 and the period in step 4. In German, in contrast, hat is realized in step 2 and gelesen_. in step 4. The common content is as follows:

16.4.3 Proplet representation of the content

noun: John	verb: read	noun: book
fnc: read	arg: John book	fnc: read
prn: 79	prn: 79	prn:79

Deeper differences between languages arise with disparities in lexicalization and grammaticalized properties such as gender, number, animateness, honorifics, and the associated agreement. Consider the following examples (Hutchins 1986, FoCL Sects. 2.4, 2.5):

16.4.4 Difficulties for achieving translation equivalence

1. The men killed the women. Three days later they were caught.
   The men killed the women. Three days later they were buried.
2. know wissen savoir
   kennen connaître
3. The watch included two new recruits that night.
   is slow.

When translating examples 1 into French, it must be decided whether they should be mapped into ils or elles. The disambiguation required is caused by a language-dependent difference in the coding of content. Plural nouns in English have neither grammatical nor natural gender, as reflected by the following partial proplet analysis of the content:

16.4.5 Content in English source language

$$\begin{bmatrix}\text{noun: man}\\ \text{cat: pnp}\\ \text{sem: def pl}\\ \ldots\\ \text{prn: 367}\end{bmatrix} \ldots \begin{bmatrix}\text{noun: woman}\\ \text{cat: pnp}\\ \text{sem: def pl}\\ \ldots\\ \text{prn: 367}\end{bmatrix} \ldots \begin{bmatrix}\text{noun: pro3}\\ \text{cat: pnp}\\ \text{sem: pl}\\ \ldots\\ \text{prn: 368}\end{bmatrix} <$$

The pronoun they agrees with two compatible antecedents, i.e. the men and the women, leaving the correct coreferential interpretation of pro3 to the hearer's pragmatic interpretation (inferencing on world knowledge).

In French, in contrast, plural nouns and pronouns are marked for grammatical gender. This is reflected in a coding of content which differs slightly from the English counterpart 16.4.5 (see <):

16.4.6 Content in French target language

$$\begin{bmatrix}\text{noun: man}\\ \text{cat: pnp}\\ \text{sem: def pl m}\\ \ldots\\ \text{prn: 367}\end{bmatrix} \ldots \begin{bmatrix}\text{noun: woman}\\ \text{cat: pnp}\\ \text{sem: def pl f}\\ \ldots\\ \text{prn: 367}\end{bmatrix} \ldots \begin{bmatrix}\text{noun: pro3}\\ \text{cat: pnp}\\ \text{sem: pl f}\\ \ldots\\ \text{prn: 368}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: (woman 367)}\\ \text{cat: pnp}\\ \text{sem: pl f}\\ \ldots\\ \text{prn: 368}\end{bmatrix} <$$

Because a pronoun and its antecedent must agree, there is only one coreferential interpretation. Depending on the pronoun, i.e. ils vs. elles, the DBS mt system may code coreference by replacing the value pro3 with the antecedent's address, here (woman 367),[9] without having to rely on world knowledge.

Example 2 of 16.4.4 illustrates the phenomenon of a *lexical gap*: whereas French and German distinguish between savoir–wissen and connaître–kennen, English provides only one word, know. Therefore a translation from English into French or German makes it necessary to choose the word which is correct for the target language. Example 3 shows a language-specific lexical homonymy: the translator must decide whether watch should be mapped into guard or clock in the target language.

Such divergences between natural languages raise the following question:

16.4.7 The interlingua question for machine translation

> Should all languages map into and out of the same universal content representation (interlingua), or should different languages use moderately language-dependent content representations (Sect. 3.6) for which point-by-point equivalences may be established when needed?

---

[9] NLC (2006) followed generative grammar by using subscripts for asserting identity, e.g. $\text{man}_i \ldots \text{he}_i$. The alternative coding of coreference by means of addresses uses the general method of establishing semantic relations of structure in DBS, thus avoiding the introduction of a separate mechanism.

If it were theoretically and practically feasible, the use of an interlingua would be ideal for machine translation. The question is how it should be constructed. Should it be enriched, for example with gender distinctions in plural nouns to accommodate translation from English to French? A similar dilemma arises if a meaning is expressed by a phrase in one language, for example, English beat a hasty retreat, and a single word in another, for example, German abhauen: should the universal interlingua treat their common meaning in the syntax or the morphology? Neither can be motivated very well (embarrassment of riches, FoCL Sect. 22.2).

Given that monolingual communication is the norm and translation between languages is the exception, running the cycle of natural language communication through an interlingua is not practical.[10] Instead, differences between languages which go beyond the standardizations of content listed in 16.4.1 need to be patched case by case, based on a careful linguistic analysis of the natural languages in question and an expert knowledge of the domain.

While the adequate recoding of a clear language content in another language is mostly a matter of effort, a translation of poetry cannot be more than an approximation. Apart from the difficulty of knowing what the author meant *exactly*, there are the differences between the word form surfaces of the source and the target language. Consider the song text by Wayne Carson, first performed by The Box Tops and released in 1967:

> Give me a ticket for an aeroplane
> Ain't got time to take a fast train
> Lonely days are gone, I'm a goin' home
> My baby just wrote me a letter

It is impossible, even for the best of translators, to adequately replicate these four lines in French. For example, the translation for aeroplane/fast_train, namely avion/rapide, does not rhyme at all.

16.5 Modeling Natural Language Understanding

The difficulties of achieving content equivalence in simultaneous interpretation and in translation stem ultimately from the way in which monolingual communication works. The mechanism functions well if the transmission of

[10] Past commercial efforts have shown that even software systems for translating between only two languages sooner or later drown in a quagmire of exceptions and special cases, such that the output for significant documents must always be checked by someone fluent in the source and the target language.

the language surfaces is loud and clear, and if speaker and hearer are familiar with the language, with each other, with their respective utterance and interpretation situations, and with the domain of the content to be transmitted.

However, even under favorable conditions a speaker may not be certain of being understood correctly and completely, just as a hearer may not be certain of how well (s)he understood the speaker. This uncertainty depends on how much use is made of the expressive power of natural language.

Explaining something new requires creative language use, but comes with a low degree of certainty. This is shown by teaching. Increasing certainty takes time: if the student can describe what is unclear, the teacher may provide additional explanation (turn taking). Leaving limits on the teacher's wisdom and the student's attention span aside, this may go on indefinitely.

Well-ingrained routines, in contrast, provide a high degree of certainty, but neither need nor allow for the use of expressive power. This is shown by situations in which fast, unambiguous content transmission is of the essence, for example, in an emergency room. Here *elaborated code* or *restricted code* (Bernstein 1971, 1973) is used: the team members have acquired the same background knowledge, mastered the same special, domain-dependent conventions of terminology, and repeat the same clearly defined procedures.

A good way to study the conditions which decide success or failure of natural language communication is analyzing recordings of natural dialogs, as in conversation analysis, a sociolinguistic approach founded by Sacks and Schegloff in the 1960s. The Schegloff (2007) corpus provides many authentic examples of brief conversations, all with audio and many with video recordings.

These conversations may be reconstructed computationally[11] by modeling (i) the utterance situation as part of the speaker's cognition, (ii) the content transported between the communicating partners, (iii) the mechanism used by the speaker to map the content into language, and (iv) the mechanism used by the hearer to reconstruct the content from the language sign (4.1.2). The method may be used also for the study of other kinds of communication, e.g. child language acquisition based on the CHILDES corpus (MacWhinney 1991).

It seems advisable to implement the communication of simple contents coded with literally used word meanings and a plain syntax before turning to nonliteral uses. The analysis of simple contents includes not only the speak and the hear mode, but also the associated nonlanguage recognition and action as well as the inferencing.

---

[11] The facts observed from a sociolinguistic point of view are well documented and carefully analyzed, but a computational reconstruction is not one of Schegloff's goals. A DBS interpretation of the Schegloff corpus will have to translate such founding notions as "pre," "post," "pre-pre," etc., (time-linear!) into DBS inferencing for maintaining the partners' state of balance.

For example, adequately reporting The Eagle has landed. (statement in the speak mode) requires the agent not only to know the referent of Eagle and to have a concept for landing, but to use them for recognizing the nonlanguage content in the first place. Understanding and fulfilling Open the door! (command in the hear mode) requires concepts which are used not only for language understanding, but also for nonlanguage recognition (door) and nonlanguage action (open). Understanding and appropriately answering What time is it? (question in the hear mode) requires a concept of time for understanding the question and of time specification for the answer (language action).

The reconstruction of concrete communication examples in DBS allows to directly observe, evaluate, and adjust the robot's cognitive operations via the service channel (Sect. 2.4; NLC Sect 1.4). This powerful method corresponds to what Norman (1981) has called *cognitive engineering*.

Once dialogs with literal language use have been mastered, more ambitious tasks may be considered. For example, one might attempt a reconstruction of register[12] adaptation, i.e. the agent's ability to adjust to a formal or informal setting, to choose the right honorifics and the right degree of politeness and familiarity, to smoothly agree with the partner in discourse on a certain level of abstraction, to move the level of abstraction up or down, to control dialect and intonation, to select between a declarative, interrogative, or imperative sentential mood, and to present content in a certain way.[13]

As a cognitive ability, register adaptation goes beyond the transmission of simple contents with literally used word forms. It requires a model of the motivational structures behind the actions of the partners in discourse (subjunctive transfer, Sect. 5.6) and the ability to devise compensatory strategies for maintaining or regaining the own state of balance. The computational reconstruction of this ability is not only of theoretical interest, but also serves to directly improve the talking robot's conversation skills as speaker and as hearer, making it more user-friendly.

16.6 Typology of DBS Operations

DBS operations include the (i) language-dependent concatenation in the hear mode, (ii) the selective activation and (iii) the inferencing of the think mode, and (iv) the language-dependent surface realization of the speak mode. The

[12] Following Halliday and Hasan (1976), the term register is widely used to mean the channel of communication from a social point of view.

[13] Register adaptation is part of constructing a correct *discourse model*.

basic distinctions between these DBS operations are (a) the number of pattern proplets and (b) the kinds of interaction between them.

The operation kinds of the hear, activation, inference, and speak modes have in common that (i) they are driven by the same motor[14] and (ii) are based on a matching between a pattern and an input, thereby binding constants to variables, which enables the derivation of an output. Applying these order-free operations in a time-linear derivation order is controlled (i) by the availability of suitable input and (ii) the overall goal of maintaining the agent in a state of balance (autonomous self-organization). Let us conclude by comparing the differences and similarities of the DBS operation kinds as they apply to input.

A hear mode operation combines the next word proplet provided by automatic word form recognition with a proplet currently available at the now front. This is reflected by the nfp+nwp structure of the operation names, whereby nfp stands for 'now front proplet' and nwp for 'next word proplet'. There are two kinds, consisting of two+two and two+one proplet patterns. As an example of a two+two hear mode operation consider NOM×FV. The operation name characterizes the input to the hear mode combination as a (i) a nominative noun available at the now front and (ii) a finite verb provided by automatic word form recognition (lexical next word).

16.6.1 Two plus two patterns in hear mode (NLC 11.6.1)

NOM×FV

$$\begin{array}{l} \textit{pattern} \\ \textit{level} \end{array} \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: N}\\ \text{fnc:}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{cat: N}'\text{ X v}\\ \text{arg:}\\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: N}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{cat: \#N}'\text{ X v}\\ \text{arg: }\alpha\\ \text{prn: K}\end{bmatrix}$$

N ϵ {snp, pnp , s3, p3}; N′ ϵ {n′,ns3′, n-s3′}.
If N ϵ {s3, snp} then N′ ϵ {ns3′, n′}; otherwise, N′ ϵ {n-s3′, n′}.

$\Uparrow$ $\Downarrow$

$$\begin{array}{l} \textit{content} \\ \textit{level} \end{array} \begin{bmatrix}\text{sur: Julia}\\ \text{noun: Julia}\\ \text{cat: snp}\\ \text{sem: nm f}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix} \begin{bmatrix}\text{sur: sleeps}\\ \text{verb: sleep}\\ \text{cat: ns3}'\text{ v}\\ \text{sem: pres}\\ \text{arg:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix} \qquad \begin{bmatrix}\text{sur:}\\ \text{noun: Julia}\\ \text{cat: snp}\\ \text{sem: nm f}\\ \text{fnc: sleep}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{verb: sleep}\\ \text{cat: \#ns3}'\text{ v}\\ \text{sem: pres}\\ \text{arg: Julia}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix}$$

[14] This was first explored by using the same motor for the time-linear composition in English morphology and syntax (FoCL 16.1.1).

While the software *drivers* of computer science are usually dependent on the hardware and the operating system, this is not the case with the abstract DBS motor. The motor kicks into action whenever there is suitable input and stops when input ceases to come in. The first motor of LA grammar is defined in NEWCAT A.1.9. The Lisp code is 28 lines long.

The operation is an instance of cross-copying (slant duplex, 16.6.7): by assigning the constant Julia to the variable α and the constant sleep to the variable β in the input pattern and using the same variables in other slots of the output pattern, the semantic relation of subject/predicate is established in the output proplets at the language level. Linguistic requirements such as agreement are formulated as restrictions on the variables regarding the values which may be bound to them. These are written at the matching boundary, i.e. directly below the operation patterns.

Besides (1) cross-copying there are the hear mode operation kinds of (2) substitution, (3) absorption, and (4) suspension (NLC 3.6.3). While cross-copying has a two-proplet output, the latter combine the two input proplets into a single output proplet. Consider an instance of absorption (flat simplex, 16.6.7):

16.6.2 Two plus one pattern operation in hear mode (NLC 13.2.9)

S∪IP

pattern level

$$\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: \#X VT}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: V\_n}\\ \text{cat: VT}'\text{ SM}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: }\alpha\\ \text{cat: \#X SM ip+}\\ \text{prn: K}\end{bmatrix} \quad <$$

VT ϵ {v, vi, vimp} and SM ϵ {decl, interrog, impv}.
If VT = vi, then SM = interrog; if VT = vimp, then SM = impv.

$$\Uparrow \qquad\qquad\qquad\qquad \Downarrow$$

content level

$$\begin{bmatrix}\text{sur:}\\ \text{verb: sleep}\\ \text{cat: \#ns3}'\text{ v}\\ \text{sem: pres}\\ \text{arg: Julia}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix}_2 \begin{bmatrix}\text{sur: }\bullet\\ \text{verb: v\_1}\\ \text{cat: v}'\text{ decl}\\ \text{prn:}\end{bmatrix}_3 \qquad \begin{bmatrix}\text{sur:}\\ \text{verb: sleep}\\ \text{cat: \#ns3}'\text{ decl ip+}\\ \text{sem: pres}\\ \text{arg: Julia}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 1}\end{bmatrix}_2 \quad <$$

As a function word, the period is absorbed into the verb proplet, replacing the cat value v with decl. The application of S∪IP triggers a body flush (NLC Sect. 11.3), so that the only proplet remaining at the now front is the current top verb. It is needed for a possible extrapropositional V−V coordination (NLC 11.6.5, 13.3.8, 13.4.10) between two propositions in a text. In contradistinction to nonlanguage recognition (Chap. 8, Sect. 13.6), language recognition (hear mode) is based on the identification of monomodal language surfaces.

Next let us turn to the think mode. It uses (i) DBS.Nav for selective activation by navigating along existing semantic relations between proplets and (ii) DBS.Inf for deriving new content from activated content by means of inferences.

DBS.Nav operations take a single proplet stored in the agent's word bank as input and activate a single proplet, also stored in the word bank, as output.

This is in contradistinction to DBS.Hear operations, which take two proplets as input and produce two or one proplet as output.

DBS.Nav operations are named according to the convention cap sr nap, where cap stands for 'currently activated proplet', sr for 'semantic relation', and nap for 'next activated proplet'. For example, the operation V$\swarrow$N takes an activated verb proplet V (here *hit* as input and uses the subject/predicate relation to access and activate the associated subject noun N (here *car*):

16.6.3 ONE PLUS ONE PATTERN IN SELECTIVE ACTIVATION (NLC 14.2.2)

$$
\begin{array}{lllll}
 & \textbf{V}\swarrow\textbf{N} & & & \\
\textit{pattern level} & \begin{bmatrix}\text{verb: } \alpha \\ \text{arg: } \beta \text{ Y} \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{noun: } \beta \\ \text{fnc: } \alpha \\ \text{prn: K}\end{bmatrix} & \text{\#-mark } \beta \text{ in the arg slot of proplet } \alpha \\
 & \Uparrow & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{sur:} \\ \text{verb: hit} \\ \text{cat: \#n}' \text{ \#a}' \text{ decl} \\ \text{sem: past} \\ \text{arg: car tree} \\ \text{mdr:} \\ \text{nc: (speed 5)} \\ \text{pc:} \\ \text{prn: 4}\end{bmatrix} & & \begin{bmatrix}\text{sur:} \\ \text{noun: car} \\ \text{cat: snp} \\ \text{sem: def sg} \\ \text{fnc:} \\ \text{mdr: heavy\&} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 4}\end{bmatrix} &
\end{array}
$$

The patterns of a DBS.Nav operation may be complemented by (i) an *instruction*, shown to the right, and (ii) a *condition* written at the matching boundary directly below the operation patterns (see next example).

A selective activation may be turned into a language-dependent DBS.Speak operation by embedding a lexicalization rule into the sur slot of the output. Consider N$\nearrow$V as it returns to the predicate and realizes the verb:

16.6.4 ONE PLUS ONE PATTERN IN SPEAK MODE (NLC 14.2.7)

$$
\begin{array}{lllll}
 & \textbf{N}\nearrow\textbf{V} & & & \\
\textit{pattern level} & \begin{bmatrix}\text{noun: } \beta \\ \text{fnc: } \alpha \\ \text{mdr: Z} \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{sur: lexverb}(\hat{\alpha}) \\ \text{verb: } \alpha \\ \text{arg: \#}\beta \text{ Y} \\ \text{prn: K}\end{bmatrix} & \text{\#-mark } \alpha \text{ in the fnc slot of proplet } \beta \\
 & \text{where Z is \#-marked or NIL} & & & \\
 & \Uparrow & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{sur:} \\ \text{noun: car} \\ \text{cat: snp} \\ \text{sem: def sg} \\ \text{fnc: hit} \\ \text{mdr: \#heavy\&} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 4}\end{bmatrix} & & \begin{bmatrix}\text{sur: hit} \\ \text{verb: hit} \\ \text{cat: \#n}' \text{ \#a}' \text{ decl} \\ \text{sem: past} \\ \text{arg: \#car tree} \\ \text{mdr:} \\ \text{nc: (speed 5)} \\ \text{pc:} \\ \text{prn: 4}\end{bmatrix} &
\end{array}
$$

The three lexicalization rules for English are called lexverb (NLC 12.4.3), lexnoun (NLC 12.5.2), and lexadj (NLC 12.6.1). The argument $\hat{\alpha}$, pronounced hat alpha, is a language-dependent counterpart of the proplet's core value, e.g. French heurter for the core value hit.

The second kind of think mode operation besides the selective activation of DBS.Nav is inferencing (Sect. 13.5). In contradistinction to the other kinds of DBS operations, inferences are unrestricted regarding the number of input and output patterns. For example, 5.3.1 has one input and one output pattern, 5.3.5 has three input and two output patterns, and 13.5.1, repeated below, has two input and two output patterns:

16.6.5 Two plus two patterns in inference (forward chaining)

$$
\begin{array}{llllll}
 & \multicolumn{2}{c}{\textit{antecedent}} & & \multicolumn{2}{c}{\textit{consequent}} \\
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: K}\end{bmatrix} & \textbf{caus} & \begin{bmatrix}\text{noun: Fido}\\ \text{fnc: bark}\\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: Fido}\\ \text{prn: K+1}\end{bmatrix} \\
\textit{matching and binding} & & \Uparrow & & & \Downarrow \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: 22}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: 22}\end{bmatrix} & & \begin{bmatrix}\text{noun: Fido}\\ \text{fnc: bark}\\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: Fido}\\ \text{prn: 23}\end{bmatrix} \\
 & \multicolumn{2}{c}{\textit{input}} & & \multicolumn{2}{c}{\textit{output}}
\end{array}
$$

The variable number of input and output patterns is complemented by a variety of connectives such as impl(ly), caus(e), sum(marize), prom(ise), exec(ute), inst(antiates), and cm (countermeasure), which have not yet consolidated into a final list. This stands in the way of a standardized naming convention for inferences, in contradistinction to the naming of the DBS.Hear and DBS.Nav operations.

In addition to classifying the DBS hear mode operations according to the number of input and output patterns, they may be classified in terms of the copying they use:

16.6.6 Kinds of cross-copying by hear mode operations

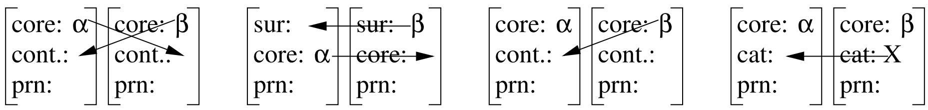
1. *duplex* vs. *simplex*
   An operation with at least two input patterns is *duplex* if (i) the core value of the one pattern is copied into a continuation slot of another and (ii) the core value of the other pattern is copied into a continuation slot of the first. An operation is *simplex* if only one of the two conditions holds.
2. *slant* vs. *flat*
   A copying operation is *slant* if a value in one proplet is copied into a dif-

ferent slot of the other and *flat* if the value is copied into the same slot of the other.

The pairs *duplex/simplex* and *slant/flat* combine into the following four constellations, which are all used in DBS:

16.6.7 THE FOUR KINDS OF CROSS-COPYING IN DBS

(i) slant duplex *(ii) flat duplex* *(iii) slant simplex* *(iv) flat simplex*



An example of (i) slant duplex is 16.6.1, an example of (ii) flat duplex is the operation of generalized baptism enabling name-based reference (HBTR 3.4.2), an example of (iii) slant simplex is shown in 7.5.12, and an example of (iv) flat simplex in 7.2.5.

In summary, the agent's hear mode operations are triggered by matching input resulting from incoming proplets provided by automatic word form recognition. In parallel, non-language recognition provides context content using the same interfaces, the same data structure, and similar operations. Concept-based reference is reconstructed cognitively as a pattern matching between the two kinds of content. The agent's think mode consists of (i) selective activation, implemented as a navigation along the semantic relations connecting stored proplets by address, and (ii) inferences deriving new content for reasoning and as blue prints for action. The language-dependent speak mode rides piggyback on the think mode. The motor driving the think mode operations is the continuous changes in the agent's ecological niche, which require countermeasures (actions) to maintain the agent in a state of balance.

Concluding Remark

In building a talking robot, two basic approaches may be roughly distinguished: bottom up and top down. The bottom up approach is hardware oriented and wrestles with mechanical aspects, as when building and controlling an artificial hand. The top down approach, in contrast, is theory-oriented and begins with the design of a general component structure, including an agent-internal memory and a functional flow from recognition to action.

This book takes a top down approach by presenting the theoretical foundation for a general model of natural language communication, suitable for free human-machine communication in real time. It includes (i) the definition of external interfaces, data structure, algorithm, and database schema; (ii) the analysis of the semantic relations of functor-argument, coordination, and coreference at the elementary, phrasal, and clausal level; (iii) the outline of an autonomous control for maintaining the robot in a continuous state of balance, based on recognition, coactivation, inferencing, and action; the handling (iv) of perspective on content in the speak and the hear mode, and (v) of reference as a cognitive interaction between the language and the context level; (vi) the explicit interpretation and production of simple as well complex[15] grammatical constructions; (vii) the production and interpretation of nonliteral language uses, and (viii) illuminating the relation between language universals and linguistic relativism.

Because the nature of cognition is a highly interdisciplinary topic, a computational reconstruction of its functions touches on numerous related issues in philosophy of language, truth-conditional semantics, traditional linguistics, language typology, nativist linguistics, corpus linguistics, natural language processing, mathematical complexity theory, and cognitive psychology. When such issues arise, they are addressed to the best of our ability. The purpose is to anchor Database Semantics in the surrounding fields of science.

[15] They include copula constructions, prepositional objects, subject sentence, object sentence, adnominal modifier sentence (a.k.a. relative clause), adverbial modifier sentence, gapping (including subject, verb, object, and noun gapping), infinitives (including bare infinitives), object sentence recursion (including unbounded dependencies), sentential mood, and verbal mood and voice.

List of Examples, Definitions, Figures, and Tables

In this text, linguistic examples, figures, definitions, tables, and summaries are contained in *risci*. Typographically, a riscus is a general purpose container, beginning with a numbered heading and ending when normal text resumes. The three part number of a heading specifies (i) the chapter and (ii) the section in which the riscus is contained as well as (iii) the number of the riscus within its section. For convenience, these headings, altogether 429, are listed below:

3. Mystery Number Two: Natural Language Communication Cycle

4. Mystery Number Three: Memory Structure

5. Mystery Number Four: Autonomous Control

6. Mystery Number Five: Learning

Part II: The Coding of Content

7. Compositional Semantics

8. Building Nonlanguage Content

9. Aspects of Language Content

10. Computing Perspective in Dialogue

11. Computing Perspective in Text

Part III: From Foundations to Applications

12. How DBS Evolved

13 Semantics

14 Natural Prototype and Artificial Model

15 Corpus Linguistics

16 Practical Benefits of Basic Research

Bibliography

Ágel, V., L. Eichinger, H.-W. Eroms, P. Hellwig, H.-J. Heringer, and H. Lobin (eds.) (2006) *Dependenz und Valenz: Ein Internationales Handbuch der zeitgenoessischen Forschung,* Vol. 2, Berlin: De Gruyter

Aho, A. V., and J. D. Ullman (1977) *Principles of Compiler Design*, Reading, MA: Addison-Wesley

AIJ = Hausser, R. (2001) "Database Semantics for Natural Language," *Artificial Intelligence*, Vol. 130.1:27–74

Ajdukiewicz, K. (1935) "Die syntaktische Konnexität," *Studia Philosophica*, Vol. 1:1–27

Anderson, J.R. (1983) "A spreading activation theory of memory," *Journal of Verbal Learning and Verbal Behavior*, Vol. 22:261–295

Anderson, J.R., and G.H. Bower (1973) *Human Associative Memory*, Washington, D.C.: V.H. Winston

Anderson, J.R., and G.H. Bower (1980) *Human Associative Memory: A Brief Edition*, Hillsdale, New Jersey: Lawrence Erlbaum Associates

Arena, P., and L.Patanè (eds.) (2009) *Spatial Temporal Patterns for Action-Oriented Perception in Roving Robots*, Cognitive Systems Monographs, Springer

Bachman, C.W. (1973) "The programmer as navigator," 1973 ACM Turing Award Lecture, *Comm. ACM,* Vol. 16.11:653–658

Bar-Hillel, Y. (1953) "Some linguistic problems connected with machine translation," *Philosophy of Science*, Vol. 20:217–225

Bar-Hillel, Y. (1964) *Language and Information. Selected Essays on Their Theory and Application*, Reading, Mass.: Addison-Wesley

Barsalou, L. (1999) "Perceptual symbol systems," *Behavioral and Brain Sciences*, Vol. 22:577–660

Barthes, R. (1986) *The Rustle of Language*, New York: Hill and Wang

Barwise, J., and J. Perry (1983) *Situations and Attitudes*, Cambridge, MA: MIT Press

Bauer, M. (2011) *An Automatic Word Form Recognition System for English in JSLIM*, M.A. thesis, CLUE (Computational Linguistics U. Erlangen)

Beauvoir, S. de (1960) *La force de l'âge*, Paris: Gallimard

ben-Zineb, R. (2010) *Implementierung eines automatischen Wortformerkennungssystems für das Arabische in JSLIM*, M.A. thesis, CLUE

Bernard. C. (1865) *Introduction à l'étude de la médecine expérimentale,* first English translation by Henry Copley Greene, published by Macmillan, 1927; reprinted 1949

Bernstein, B. (1971) *Class, Code and Control: Volume 1 - Theoretical Studies Towards A Sociology Of Language*, Routledge

Bernstein, B. (1973) *Class, Code and Control: Volume 2 - Applied Studies Towards A Sociology Of Language*, Routledge

Berwick, R.C., and A.S. Weinberg (1984) *The Grammatical Basis of Linguistic Performance: Language Use and Acquisition*, Cambridge, Mass.: MIT Press

Betti, A. (2004) "Leśniewski's early Liar, Tarski, and natural language," *Annals of Pure and Applied Logic*, Vol. 127:267–287

Beutel, B. (2009) *Malaga, Grammar Development Environment for Natural Languages*, http://home.arcor.de/bjoern-beutel/malaga/

Beynon-Davies, P. (2003) *Database Systems*, Basingstoke: Palgrave Macmillan

Bibel, W. (1993) *Wissensrepräsentation und Inferenz*, Braunschweig-Wiesbaden: Vieweg

Biederman, I. (1987) "Recognition-by-components: a theory of human image understanding," *Psychological Review*, Vol. 94:115–147

Biederman, I., and P.C. Gerhardstein (1993) "Recognizing depth-rotated objects: Evidence and conditions for three-dimensional viewpoint invariance," *Journal of Experimental Psychology: Human Perception and Performance*, Vol 19(6):1162-1182

Blair, D.C., and M.E. Maron (1985) "An evaluation of retrieval effectiveness for a full-text document-retrieval system," *Comm. ACM* 28.3:289–299

Bloomfield, L. (1933) *Language*, Holt, Rinehart, and Winston, New York.

Boas, F. (1911) *Handbook of American Indian languages (Vol. 1)*, Bureau of American Ethnology, Bulletin 40. Washington: Government Print Office (Smithsonian Institution, Bureau of American Ethnology)

Bosch, P. (1986) “Pronouns under control? A reply to Lilliane Tasmosvki and Paul Verluyten,” *Journal of Semantics*, Vol. 5:65–78

Brachman, R.J. (1979) “On the epistemological status of semantic networks,” in N. Findler (ed.), p. 3–50; Reprinted in Brachman and Levesque (eds.) 1985

Brachman, R.J., and Levesque (eds.) (1985) *Readings in Knowledge Representation*, Los Altos, CA: Morgan Kaufmann

Braitenberg, V. (1984) *Vehicles: Experiments in Synthetic Psychology*, Cambridge, MA: MIT Press

de la Briandais, R. (1959) ”File Searching Using Variable Length Keys,” Proc. Western Joint Computer Conf. 15, 295–298

Brinton, L.J., and E. Closs Traugott (2005) *Lexicalization and Language Change*, CUP

Brooks, R. (1985) “A robust layered control system for a mobile robot,” Cambridge, MA: *MIT AI Lab Memo 864*, p. 227–270

Brooks, R. (1990) “Elephants don’t play chess,” MIT Artificial Intelligence Laboratory, *Robotics and Autonomous Systems,* Vol. 6:3–15

Brooks, R. (1991) “Intelligence without representation,” *Artificial Intelligence*, Vol. 47:139–160

Brown, P., S. Della Pietra, V. Della Pietra, and R. Mercer (1991) “Word Sense Disambiguation Using Statistical Methods,” Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, CA, June 1991, 264–270

Burnard, L. (ed.) (1995) *Users Reference Guide British National Corpus Version 1.0*, Oxford U. Computing Services, England: Oxford

Burnard, L. (ed.) (2007) *Reference Guide for the British National Corpus (XML Edition)*, Oxford U. Computing Services, England: Oxford

Buszkowski, W. (1988) “Gaifman’s theorem on categorial grammars revisited,” *Studia Logica*, Vol. 47.1:23–33

Campbell, N.A. (1996) *Biology, 4th ed.*, New York: Benjamin Cummings

Carbonell, J.G., and R. Joseph (1986) “FrameKit+: a knowledge representation system,” Carnegie Mellon U., Department of Computer Science

Carpenter, B. (1992) *The Logic of Typed Feature Structures*, Cambridge: CUP

CELEX2 (1996) Olac Record, http://olac.ldc.upenn.edu/item/oai:www.ldc.upenn.edu:LDC96L14

Chen, P. (1976) "The entity-relationship model – toward a unified view of data," *ACM Transactions on Database Systems*, p. 9–36 ACM-Press

Chisvin, L., and R.J. Duckworth (1992) "Content-addressable and associative memory," p. 159–235, in M.C. Yovits (ed.)

Chomsky, N. (1957) *Syntactic Structures*, The Hague: Mouton

Chomsky, N. (1965) *Aspects of the Theory of Syntax*. Cambridge, MA : The MIT Press.

Chomsky, N. (1995) *The Minimalist Program*, Current Studies in Linguistics 28, Cambridge, MA : MIT Press,

Chomsky, N., and M. Halle (1968) *The Sound Pattern of English*, New York: Harper Row

Church, K., and R.L. Mercer (1933) "Introduction to the Special Issue on Computational Linguistics Using Large Corpora," *Computational Linguistics*, Vol. 19.1:1–24

CLaTR'11 = Hausser, R. (2011) *Computational Linguistics and Talking Robots, Processing Content in Database Semantics, 1st ed.*, pp. 286, Springer

CoL = Hausser, R. (1989) *Computation of Language, An Essay on Syntax, Semantics, and Pragmatics in Natural Man-Machine Communication*, Symbolic Computation: Artificial Intelligence, pp. 425, Springer

Cole, R. (ed.) (1997) *Survey of the State of the Art in Human Language Technology*, Edinburgh: Edinburgh U. Press

Comrie, B. (1986) "Reflections on subject and object control," *Journal of Semantics*, Vol. 4:47–65, Oxford: OUP

Copestake, A., D. Flickinger, C. Pollard, and I. Sag (2006) "Minimal recursion semantics: an introduction," *Research on Language and Computation* 3.4:281–332

Corbett, A.T., J.R. Anderson, and E.J. Patterson (1988) "Problem compilation and tutoring flexibility in the Lisp tutor," p. 423–429, Montréal: ITS-88

Cormen, T.H., C.E. Leiserson, R.L. Rivest, and C. Stein (2009) *Introduction to Algorithms, 3rd ed.*, Cambridge, MA: MIT Press

Croft, W. (2000) "Parts of speech as language universals and as language-particular categories," in Vogel and Comrie 2000, (eds.) *Approaches to the Typology of Word Classes*, Berlin: Mouton de Gruyter p. 65–102

Cuvier, G. (1817/2009) *Le règne animal; distribué d'après son organisation; pour servir de base à l'histoire naturelle des animaux et d'introduction à l'anatomie comparé,* 4 Vol., Paris

Darwin, C. (1839/1989) *The Voyage of the Beagle*, Penguin Classics

Davidson, D., and G. Harman (eds.) (1972) *Semantics of Natural Language*, Dordrecht: D. Reidel

Davies, M. (2008) *The Corpus of Contemporary American English*, Brigham Young University. http://corpus.byu.edu/coca/

DeRose, S. (1988) "Grammatical Category Disambiguation by Statistical Optimization," *Computational Linguistics*, Vol. 14.1:31–39

Dik, S. (1989) *The Theory of Functional Grammar (Part I: The Structure of the Clause)*, Berlin, New York: Mouton de Gruyter

Dik, S. (1997) *The Theory of Functional Grammar (Part II: Complex and Derived Constructions)*, Berlin, New York: Mouton de Gruyter

Dodt, E., and Y. Zotterman (1952) "The Discharge of Specific, Cold Fibres at High Temperatures," *Acta Physiologica Scandinavica*, Vol 26.4: 358–365, December 1952

Dominey, P.F., and F. Warneken (2011) "The basis of shared intentions in human and robot cognition," *New Ideas in Psychology*, Vol. 29.3:260–274, Elsevier

Donald, M. (1991) *The Origins of the Modern Human Mind*, Cambridge, MA: Cambridge University Press

Dorr, B.J. (1993) *Machine Translation: A View from the Lexicon*, Cambridge, MA: MIT Press

Ekman, P. (2003) "16 Enjoyable Emotions," *Emotion Researcher*, Vol. 18:6–7

Elmasri, R., and S.B. Navathe (2010) *Fundamentals of Database Systems, 6th ed.*. Redwood City, CA: Benjamin-Cummings

Endsley, M., and D.J. Garland (eds.) (2000) *Situation Awareness and Measurement*, Mahwah, New Jersey: Lawrence

Endsley, M., B. Bolte, and D.G. Jones (2003) *Designing for Situation Awareness*, London and New York: Taylor and Francis

Everett, D. (2005) "Cultural Constraints on Grammar and Cognition in Pirahã: Another Look at the Design Features of Human Language," *Current Anthropology* Vol 46:621–646

Everett, D. (2013) *Language: The Cultural Tool*, London: Profile Books

Fikes, R.E., and N. Nilsson (1971) "STRIPS: a new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, Vol. 2:189–208

Fillmore, C. (1968) "The Case for Case". In Bach and Harms (eds.): *Universals in Linguistic Theory*, New York: Holt, Rinehart, and Winston, 1-88

Fillmore, C., P. Kay, and C. O'Connor (1988) "Regularity and idiomaticity in grammatical constructions: the case of let alone," *Language*, Vol. 64: 501–38

Findler. N. (ed.) (1985) *Associative Networks*, Academic Press

Fischer, W. (2002) *Implementing Database Semantics as an RDMS* (in German), Studienarbeit am Institut für Informatik der Universität Erlangen-Nürnberg (Prof. Meyer-Wegener), published as CLUE-Arbeitsbericht 7 (2004). http://www.linguistik.uni-erlangen.de/clue/de/arbeiten/arbeitsberichte.html

FoCL = Hausser, R. (1999) *Foundations of Computational Linguistics, Human-Computer Communication in Natural Language, 3rd ed. 2013*, Springer

Fornwald, S. (2012) *Implementierung eines Parsers für elementare, phrasale und clausale Nomina im Georgischen*, M.A. thesis, CLUE

Fowler, M. (2003) *UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed.*, Addison-Wesley

Francis, W.N., and H. Kučera (1982) *Frequency Analysis of English Usage: Lexicon and Grammar*, Boston: Houghton Mifflin

Fredkin, E. (1960) "Trie memory," *Comm. ACM*, Vol. 3.9:490–499

Frege, G. (1879) *Begriffsschrift. Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle: L. Nebert

Frege, G. (1891) *Function und Begriff*, Jena: Hermann Pohle

Frege, G. (1892) "Über Sinn und Bedeutung," *Zeitschrift für Philosophie und philosophische Kritik*, Vol 100: 25–50

Frege, G. (1967) *Kleine Schriften*, I. Angelelli (ed.), Darmstadt: Wissenschaftliche Buchgesellschaft

Gao, M. (2007) *An Automatic Word Form Recognition System for German in Java*, M.A. thesis, CLUE

Garside, R., G. Leech, et al. (1987) *The Computational Analysis of English*, London: Longman

Geach P. (1962) *Reference and Generality: An Examination of Some Medieval and Modern Theories*, Ithaca, New York: Cornell University Press

Geach, P. (1969) "A program for syntax," in D. Davidson and G. Harman (eds.), p. 483–497

Geurts, B. (2002) "Donkey business," *Linguistics and Philosophy*, Vol. 25:129–156

Girstl, A. (2006) *Development of Allomorph-Rules for German Morphology Within the Framework of JSLIM,* B.A. thesis, CLUE

Givón, T. (1990) *Syntax: A Functional Typological Introduction* (2 Volumes), Amsterdam: John Benjamins

Graf, S. (2011) *Implementierung eines Chinesischen Lexikons in JSLIM*, M.A. thesis, CLUE

Greenberg, J. (1963) "Some universals of grammar with particular reference to the order of meaningful elements," in J. Greenberg (ed.) *Universals of Language*. Cambridge, MA: MIT Press

Greiner, P. (2011) *Syntaktische Analyse des Englischen mit JSLIM*, M.A. thesis, CLUE

Grice, P. (1957) "Meaning," *Philosophical Review*, Vol. 66:377–388

Grice, P. (1965) "Utterer's meaning, sentence meaning, and word meaning," *Foundations of Language*, Vol. 4:1–18

Grice, P. (1969) "Utterer's meaning and intention," *Philosophical Review*, Vol. 78:147–77

Groenendijk, J.A.G., T.M.V. Janssen, and M.B.J. Stokhof (eds) (1980) *Formal Methods in the Study of Language*, U. of Amsterdam: Mathematical Center Tracts 135

Grün, A. von der (1998) *Wort-, Morphem- und Allomorphhäufigkeit in domänenspezifischen Korpora des Deutschen*, M.A. thesis, CLUE

Hajičová, E. (2000) "Dependency-based underlying-structure tagging of a very large czech corpus," in: Special issue of TAL Journal, Grammaires de Dépendence/Dependency Grammars, p. 57–78, Paris: Hermes

Halliday, M.A.K., and R. Hasan (1976) *Cohesion in English*, London: Longman

Hammer, B., A. Micheli, A. Sperduti, and M. Strickert (2004) "Recursive self-organizing network models," *Neural Netw.* 2004 Oct-Nov;17(8-9):1061-85. http://www.researchgate.net/publication/8171170

Handl, J. (2008) *Entwurf und Implementierung einer abstrakten Maschine für die oberflächenkompositionale inkrementelle Analyse natürlicher Sprache*, Diplom thesis, Department of Computer Science, U. of Erlangen-Nürnberg

Handl, J. (2012) *Inkrementelle Oberflächenkompositionale Analyse und Generierung Natürlicher Sprache*, Inaugural Dissertation, CLUE, Univ. Erlangen–Nürnberg. http://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/3933)

Handl, J., and C. Weber (2010) "A multilayered declarative approach to cope with morphotactics and allomorphy in derivational morphology", in *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010)*, p. 2253–2256, Valletta, Malta

Harris, Z. (1951) *Method in Structural Linguistics*, Chicago: University of Chicago Press

Hauser, M.D. (1996) *The Evolution of Communication*, Cambridge, MA: MIT Press

Hauser, M., Chomsky, N., and Fitch, T. (2002) "The faculty of language: what is it, who has it, and how did it evolve?" *Science* Vol. 198:1569–79

Hausser, R. (1973) "Presuppositions and quantifiers," *Papers from the Ninth CLS-Meeting*, Chicago: Chicago Linguistics Society

Hausser, R. (1974) "Syntax and semantics of plural," *Papers from the Tenth CLS-Meeting*, Chicago: Chicago Linguistics Society

Hausser, R. (1980) "The place of pragmatics in model-theory," in J.A.G. Groenendijk et al. (eds.)

Hausser, R. (2003) "Reconstructing propositional calculus in Database Semantics," in H. Kangassalo et al. (eds.)

Hausser, R. (2007) "Comparing the use of feature structures in nativism and in Database Semantics," in H. Jaakkola et al. (eds.)

Hausser, R. (2008) "Center fragments for upscaling and verification in Database Semantics," in Y. Kiyoki et al. (eds.)

Hausser, R. (2009) "Modeling natural language communication in Database Semantics," in M. Kirchberg and S. Link (eds.), *Proceedings of the APCCM 2009*, Australian Computer Science Inc., CRPIT, Vol. 96:17–26

Hausser, R. (2010) “Language production in Database Semantics,” in PACLIC 24, *Proceedings of the 24th Pacific Asia Conference on Language, Information, and Computation*, 4–7 November 2010, Tohoku U., Sendai, Japan

Hausser, R. (2015) “From Montague grammar to Database Semantics,” paper invited by *Language and Information*, Seoul: Korean Society for Language and Information

Helfenbein, D. (2005) *Die Behandlung pronominaler Koreferenz in der Datenbanksemantik*, M.A. thesis, CLUE

Hengeveld, K. (1992) *Non-Verbal Predication: Theory, Typology, Diachrony*, Berlin, New York: Mouton de Gruyter

Herlihy, M., and N. Shavit (2011), “On the nature of progress,” in *Principles of Distributed Systems*, Odopis 2011, LNCS, Vol. 7109/2011:313–328, Springer

Herrmann, T. (2003) “Kognitive Grundlagen der Sprachproduktion,” in G. Rickheit, Th. Herrmann, and W. Deutsch (eds.), *Psycholinguistik-Psycholinguistics. Ein internationales Handbuch.* p. 228–244, Berlin, New York: Mouton de Gruyter

Hubel, D.H., and T.N. Wiesel (1962) “Receptive fields, binocular interaction, and functional architecture in the cat’s visual cortex,” *Journal of Physiology*, Vol. 160:106–154.

Huddleston R., and G.K. Pullum (2002) *The Cambridge Grammar of the English Language*, Cambridge: CUP

Hudson, R. (2010) *An Introduction to Word Grammar*, Cambridge: CUP

Huezo, R. (2003) *Time Linear Syntax of Spanish*, M.A. thesis, CLUE

Hutchins, W.J. (1986) *Machine Translation: Past, Present, Future*, Chichester: Ellis Horwood

Ickler, T. (1994) “Zur Bedeutung der sogenannten ‚Modalpartikeln‘,” *Sprachwissenschaft,* Vol. 19:374–404

Ide, N., K. Suderman, C. Baker, R. Passonneau, and C. Fellbaum (2010) “MASC I: manually annotated sub-corpus first release,” LDC Catalog No.: LDC2010T22, ISBN: 1-58563-569-3, Release Date: Dec 20, 2010

ISO 24610-1 (2006) “Language resource management – feature structures – Part 1: feature structure representation,” TC 37/SC4

Ivanova, M. (2009) *Syntax und Semantik des Bulgarischen im Rahmen von JSLIM*, M.A. thesis, CLUE

Jaakkola, H., Y. Kiyoki, and T. Tokuda (eds.) (2007) *Information Modelling and Knowledge Bases XIX*, Amsterdam: IOS Press

Jackendoff, R.S. (1990) *Semantic Structures*, Cambridge, MA: MIT Press

Jackendoff, R.S. (1999) "Possible stages in the evolution of the language capacity," *Trends in Cognitive Science*, Vol. 3.2:272–279

Jacobson, P., and G.K. Pullum (eds.) (1982) *The Nature of Syntactic Representation*, Dordrecht: Reidel

Jaegers, S. (2010) *Entwurf und Implementierung von Grammatikregeln für die Analyse von Komposita im Rahmen der SLIM-Sprachtheorie*, M.A. thesis, CLUE

Johnson, D.E., and P.M. Postal (1980) *Arc Pair Grammar*, Princeton: Princeton U. Press

Jurafsky, D., and J.H. Martin (2009) *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics, 2nd ed.*, Prentice Hall

Kabashi, B. (2003) *Automatische Wortformerkennung für das Albanische*, M.A. thesis, CLUE

Kabashi, B. (2007) "Pronominal clitics and valency in Albanian. A Computational Linguistics Prespective and Modelling within the LAG-Framework," in: Herbst, Th., K. Götz-Votteler (Eds.) *Valency – Theoretical, Descriptive and Cognitive Issues,* p. 339–352, Berlin, New York : Mouton de Gruyter

Kager, R. (1999) *Optimality Theory*, CUP

Kalender, E. (2010) *Automatische JSLIM-Analyse der russischen Syntax im Rahmen der Datenbanksemantik*, M.A. thesis, CLUE

Kamp, J.A.W. (1980) "A theory of truth and semantic representation," in J.A.G. Groenendijk et al. (eds.).

Kamp, J.A.W., and U. Reyle (1993) *From Discourse to Logic*, Parts 1 and 2, Dordrecht: Kluwer

Kangassalo, H., et al. (eds.) (2003) *Information Modeling and Knowledge Bases XIV,* Amsterdam: IOS Press

Kapfer, J. (2010) *Inkrementelles und oberflächenkompositionales Parsen von Koordinationsellipsen*, Ph.D. dissertation, CLUE

Karsten, K.B., L.N. Andriamandimbiarisoa, S.F. Fox, and C.J. Raxworthy (2009) "Sexual selection on body size and secondary sexual characters in

2 closely related, sympatric chameleons in Madagascar," *Behavioral Ecology,* Vol. 20.5:1079–1088

Kass, R., and Finin, T. (1988) "Modeling the user in natural language systems," Computational Linguistics, 14:5–22

Katz, J., and J. Fodor (eds.) (1964) *The Structure of English*, Englewood Cliffs, NJ: Prentice Hall

Kay, M. (1984) "Functional unification grammar: A formalism for machine translation," COLING 84

Kay, M. (2004) *XPath 2.0 Programmer's Reference (Programmer to Programmer)*, Wrox Press

Kempson, R., and A. Cormack (1981) "Ambiguity and quantification," *Linguistics and Philosophy* 4:259–309

Kempson, R., W. Meyer-Viol, D. Gabbay (2001) *Dynamic Syntax: The Flow of Language Understanding*, Wiley-Blackwell

Kim, S. (2009) *Automatische Wortformerkennung für das Koreanische im Rahmen der LAG*, Ph.D. dissertation, CLUE

Kiyoki, Y., T. Tokuda, H. Jaakkola, X. Chen, N. Yoshida (eds.) (2008) *Information Modeling and Knowledge Bases XX*, Amsterdam: IOS Press

Klatzky, R.L., B. MacWhinney, and M. Behrmann (eds.) (1998) *Embodiment, Ego-Space, and Action*, New York: Psychology Press

Klepmann, M. (2008) *Designing Data-Intensive Applications*, O'Reilly

Klima, E. (1964) "Negation in English," in J. Katz and J. Fodor (eds.)

Knuth, D.E. (1998) *The Art of Computer Programming. Vol. 3, 2nd ed.* Boston: Addison-Wesley

Kohonen, T. (1988) *Self-Organization and Associative Memory, 2nd ed.*, Springer

Kosche, S. (2011) *Syntax und Semantik des Französischen im Rahmen von JSLIM*, M.A. thesis, CLUE

Kučera, H., and W.N. Francis (1967) *Computational Analysis of Present-Day English*. Providence, Rhode Island: Brown U. Press

Kycia, A. (2004) *An Implementation of Database Semantics in Java*. M.A. thesis, CLUE

L&I'05 = Hausser, R. (2005) "Memory-Based pattern completion in database semantics," *Language and Information*, Vol. 9.1:69–92, Seoul: Korean Society for Language and Information

L&I'10 = Hausser, R. (2010) "Language production based on autonomous control - a content-addressable memory for a model of cognition," *Language and Information*, Vol. 11:5–31, Seoul: Korean Society for Language and Information

Langacker, R. (1969) "Pronominalization and the chain of command," in D.A. Reibel and S.A. Shane (eds.), p. 160–186

Lee, Kiyong (2002) "A simple syntax for complex semantics," in *Language, Information, and Computation*, Proceedings of the 16th Pacific Asia Conference, p. 2–27

Lee, Kiyong (2004) "A computational treatment of some case-related problems in Korean," *Perspectives on Korean Case and Case Marking*, p. 21–56, Seoul: Taehaksa

Leech, G. (1995) "A Brief User's Guide to the Grammatical Tagging of the British National Corpus." http://www.natcorp.ox.ac.uk/docs/gramtag.html

Leidner, J. (2000) *Linksassoziative morphologische Analyse des Englischen mit stochastischer Disambiguierung*, M.A. thesis, CLUE

Lenders, W. (2012) "Wie Computer Sprache lernen," in I. Podtergera (ed.) *Schnittpunkt Slavistik. Ost und West im wissenschaftlichen Dialog. Festgabe für Helmut Keipert zum 70. Geburtstag*. Teil 1: Slavistik im Dialog - einst und jetzt. V&R unipress (Bonn University Press) Göttingen 2012, 455-471.

Leśniewski, S. (1929) "Grundzüge eines neuen Systems der Grundlagen der Mathematik," Warsaw: *Fundamenta Mathematicae*, Vol. 14:1–81

Library of Congress (05.2015) http://www.loc.gov/standards/sourcelist/date-time.html

Lieb, H.H. (1976) "Zum Verhältnis von Sprachtheorien, Grammatiktheorien und Grammatiken," in: D. Wunderlich (ed.). Wissenschaftstheorie der Linguistik, p. 200–214

Lipp, S. (2010) *Automatische Wortformerkennung für das Schwedische unter Verwendung von JSLIM*, M.A. thesis, CLUE

Liu, Haitao (2009a) "Statistical properties of Chinese semantic networks," *Chinese Science Bulletin*, Vol. 54.16:2781–2785

Liu, Haitao (2009b) *Dependency grammar: from theory to practice*, Beijing: Science Press

Liu, Haitao (2011) "Linguistic networks: metaphor or tool?" *Journal of Zhejiang University*, (Humanities and Social Science) Vol. 41.2:169–180.

MacWhinney, B. (1991) *The CHILDES project: Computational tools for analyzing talk*, Hillsdale, NJ: Lawrence Earlbaum Associates

MacWhinney, B. (1999) "The Emergence of Language from Embodiment,Â´Â´ in B. MacWhinney (ed.) *The Emergence of Language*, Mahwah, NJ: Lawrence Erlbaum

MacWhinney, B. (2004) "Multiple solutions to the logical problem of language acquisition," *Journal of Child Language,* Vol. 31:919–922. Cambridge U. Press

MacWhinney, B. (2008) "How mental models encode embodied linguistic perspective," in L. Klatzky et al. (eds.)

MacWhinney, B. (ed.) (1999) *The Emergence of Language from Embodiment*, Hillsdale, NJ: Lawrence Erlbaum

März, B. (2005) *Die Koordination in der Datenbanksemantik*, M.A. thesis, CLUE

Mahlow, K. (2000) *Automatische Wortformanalyse für das Spanische*, M.A. thesis, CLUE

Mahlow, K., and M. Piotrowski (eds.) (2009) *State of the Art in Computational Morphology*, Springer

Marty, A. (1908) *Untersuchungen zur Grundlegung der allgemeinen Grammatik und Sprachphilosophie*, Vol. 1, Halle: Niemeyer, reprint Hildesheim, New York: Olms (1976)

McCarthy, C. (2006) *The Road*, New York: Knopf Doubleday

McKeown, K. (1985) *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, Cambridge: CUP

Mehlhaff, J. (2007) *Funktor-Argument-Struktur und Koordination im Deutschen – eine Implementation in JSLIM*, M.A. thesis, CLUE

Mei, Hua (2007) *Automatische Syntax und Semantikanalyse des Chinesischen in Rahmen von JSLIM*, M.A. thesis, CLUE

Miller, G. (1995) "Wordnet: a lexical database," *Communications of the ACM*, Vol. 38.11:39–41

Minsky, M. (1975) "A framework for representing knowledge," in P. Winston (ed.), p. 211–277

Mitchell, T. (1997) *Machine Learning*, McGraw-Hill

Mithun, M. (1999) *The Languages of Native North America*. Cambridge U. Press

Mohammadzadeh, A., S. Soltan, and M. Rahgozar (2006), "Template guided association rule mining from XML documents," in *Proceedings of the 15th International Conference on World Wide Web*, p. 963–964, New York: ACM Press

Montague, R. (1974) *Formal Philosophy*, New Haven: Yale U. Press

Muñoz-Avila, H., U. Jaidee , D.W. Aha, and E. Carter (2010) "Goal-driven autonomy with case-based reasoning," in Proceedings of ICCBR2010, p. 228–241

Naphade, M.R., and J.R. Smith (2009) "Computer program product and system for autonomous classification," Patent Application #:20090037358 - Class: 706 46 (USPTO)

Nevins, I. A., D. Pesetsky, and C. Rodrigues (2009) "Evidence and argumentation: A reply to Everett (2009)" *Language*, Vol. 85.3: 671–681

Newell, A., and H.A. Simon (1972) *Human Problem Solving*, Englewood Cliffs, New Jersey: Prentice-Hall

Nichols, J. (1992) *Linguistic Diversity in Space and Time*, The U. of Chicago Press

Niedobijczuk, K. (2010) *Implementierung eines automatischen Wortformerkennungssystems des Polnischen in JSLIM*, M.A. thesis, CLUE

NLC = Hausser, R. (2006) *A Computational Model of Natural Language Commmunication: Interpretation, Inference, and Production in Database Semantics*, Springer

Norman, D.A., (1981) "Steps toward cognitive engineering: System images, system friendliness, mental models," San Diego, California, UCSD, Department of Psychology and Program in Cognitive Science, draft

Oberhofer, T. (1993) *Linksassoziative Analyse des Chinesischen*, M.A. thesis, CLUE

Odrzywolski, M. (2012) *The Polish Language and the Implementation of its Basic Syntactic Structures in JSLIM*, M.A. thesis, CLUE

Oostdijk, N. (1988) "A corpus linguistic approach to linguistic variation," in G. Dixon (ed.): *Literary and Linguistic Computing*, Vol. 3.1

Pakula, A.J. (1976) *All the President's Men*, Warner Brothers

Pandea, P. (2010) *Implementierung eines automatischen Wortformerkennungssystems im Rumänischen mit JSLIM*, M.A. thesis, CLUE

Pankovets, T. (2011) *Struktur und Koordination im Japanischen – eine Implementation in JSLIM*, M.A. thesis, CLUE

Peirce, C.S. *Collected Papers*. C. Hartshorne and P. Weiss (eds.), Cambridge, MA: Harvard Univ. Press. 1931–1935

Pelletier, F.J. (2001) "Did Frege Believe Frege's Principle?" *Journal of Logic, Language, and Information*, Vol. 10:87-114

Pepiuk, S. (2010) *Implementierung eines automatischen Wortformerkennungssystems für das Französische in JSLIM*, M.A. thesis, CLUE

Perlmutter, D.M. (1980) "Relational grammar," In E.A. Moravcsik and J.R. Wirth (Eds.), *Syntax and Semantics: Current Approaches to Syntax,* Vol. 13:195–229. New York: Academic Press

Peters, S., and R. Ritchie (1973) "On the generative power of transformational grammar," *Information and Control*, Vol. 18:483–501

Pollard, C., and I. Sag (1987) *Information-based syntax and semantics*, Vol. I, Fundamentals. Stanford U.: CSLI Lecture Notes 13

Proust, M. (1913) *Du côté de chez Swann*, ed. by Jean-Yves Tadie et al., Bibliothèque de la Pléiade, Paris: Gallimard, 1987–1989

PTQ = Montague, R. (1974) "The proper treatment of quantification in ordinary English," p. 247–270 in *Formal Philosophy*, New Haven: Yale U. Press

Pullum, G.K. (1991) *The Great Eskimo Vocabulary Hoax: And Other Irreverent Essays on the Study of Language*, The U. of Chicago Press

Pustejovsky, J. (1995) *The Generative Lexicon,* Cambridge, MA: MIT Press

Quillian, M. (1968) "Semantic memory," in M. Minsky (ed.), *Semantic Information Processing*, p. 227–270, Cambridge, MA: MIT Press

Quine, W.v.O. (1960) *Word and Object*. Cambridge, MA: MIT Press

Quirk R., S. Greenbaum, G. Leech, and J. Svartvik (1985) *A Comprehensive Grammar of the English Language*, London New York: Longman

Reibel, D.A., and S.A. Shane (eds.) (1969) *Modern Studies of English*, Englewood Cliffs, New Jersey: Prentice Hall

Reihl, S. (2010) *Inkrementelle und oberflächenkompositionale Analyse von Derivaten im Deutschen*, M.A. thesis, CLUE

Reis, M., and W. Sternefeld (2004) "Review of S. Wurmbrand: Infinitives. Restructuring and Clause Structure," *Linguistics*, Vol. 42:469–508

Reimer, M., and E. Michaelson (2014) "Reference", *The Stanford Encyclopedia of Philosophy* (Winter 2014 Edition), Edward N. Zalta (ed.). http://plato.stanford.edu/archives/win2014/entries/reference/

Rijkhoff, J. (2002) *The Noun Phrase*, New York, N.Y.: OUP

Rosch, E. (1999) "Reclaiming concepts", Journal of Consciousness Studies, Vol. 6.11–12:61–77

Ross, J.R. (1969) "On the cyclic nature of English pronominalization," in D.A. Reibel and S.A. Schane (eds.), p. 187–200

Roy, D. (2003) "Grounded spoken language acquisition: experiments in word learning," *IEEE Transactions on Multimedia*, Vol. 5.2:197–209

Roy, D. (2005) "Grounding words in perception and action: computational insights," *TRENDS in Cognitive Sciences*, Vol. 9.8:389–396

Roy, D. (2008) "A mechanistic model of three facets of meaning. Symbols, embodiment, and meaning," in de Vega, Glenberg, and Graesser (eds.) *Symbols and Embodiment: Debates on Meaning and Cognition*, New York, N.Y.: OUP

Rudat, S. (2011) *Implementierung eines Systems der automatischen Wortformerkennung des Portugiesischen mit dem Programm JSLIM*, M.A. thesis, CLUE

Sabeva, S. (2010) *Erstellung eines linksassoziativen morphologischen Analysesystems für das Bulgarische*, Ph.D. dissertation, CLUE

Sacks, H., Schegloff, E., and Jefferson, G. (1974) "A simplest systematics for the organisation of turn-taking for conversation," in *Language*, 50:696–735.

Sacks, H., and E. Schegloff (1979) "Two preferences in the organization of reference to persons in conversation and their interaction," in G. Psathas (ed.) *Everyday Language: Studies in Ethnomethodology*, p. 15–21, New York: Irvington Press

Sapir, E. (1921) *Language, an Introduction to the Study of Speech*, New York: Harvest Books – Harcourt, Brace, and World

Sasse, H.J. (1993) "Das Nomen – eine universelle Kategorie?" in *Sprachtypologie und Universalienforschung*, Vol. 46,3:187–221

Saussure, F. de (1916/1972) *Cours de linguistique générale*, Édition critique préparée par Tullio de Mauro, Paris: Éditions Payot

SCG = Hausser, R. (1984) *Surface Compositional Grammar*, Munich: Wilhelm Fink Verlag

Schank, R. (1982) *Dynamic Memory: A Theory of Learning in Computers and People,* New York: Cambridge U. Press

Schegloff, E.A. (2007) *Sequence Organization in Interaction,* New York: Cambridge U. Press

Schewe, K.D., and B. Thalheim (1994) "Achieving consistency in active databases," in S. Chakravarthy and J. Widom (eds.) Proc. IEEE RIDE-ADS

Schneider, T. (2011) *Implementierung eines Syntaxparsers für die Analyse syntaktisch trennbarer Verben im Deutschen*, M.A. thesis, CLUE

Schnelle, H. (1988) "Ansätze zur prozessualen Linguistik," in: Schnelle, H., and G. Rickheit (eds.), p. 137–190

Schnelle, H., and G. Rickheit (eds.) (1988) *Sprache in Mensch und Computer*, Opladen: Westdeutscher Verlag

Schörner, S. (2007) *Kategorisierung Deutscher Funktionswörter und ihre Kategorisierung in der Datenbanksemantik*, M.A. thesis, CLUE

Sells, P. (1985) *Lectures on Contemporary Syntactic Theory: an introduction to government-binding theory, generalized phrase structure grammar, and lexical-functional grammar*, Stanford: CSLI Lecture Notes Number 3

Seneschal, H.-Y. (2011) *The Grammatical Analysis of Three Particles in Modern Chinese in JSLIM*, M.A. thesis, CLUE

Shannon, C.E., and W. Weaver (1949) *The Mathematical Theory of Communication*, Urbana, Illinois: University of Illinois Press

Sharman, R. (1990) *Hidden Markov model methods for word tagging*, Report 214, IBM UK Scientific Centre, Winchester

Sinclair, J. (1991) *Corpus Concordance Collocation*, Oxford: OUP

Söllch, G. (2009) *Syntaktisch-semantische Analyse des Tagalog im Rahmen der Datenbanksemantik*, M.A. thesis, CLUE

Solé, R.V., B. Corominas-Murtra, S. Valverde, and L. Steels (2010) "Language networks: their structure, function, and evolution," *Complexity*, Vol. 15.6:20–26

Sorensen, M.N., (1997) "Infinitives (and more)," http://faculty.washington.edu/marynell/grammar/infini.html

Sowa, J. (1987/1992) "Semantic networks," *Encyclopedia of Artificial Intelligence*, edited by S.C. Shapiro; revised and extended for the second edition, 1992. New York: Wiley. http://www.jfsowa.com/pubs/semnet.htm

Steels, L. (1999) *The Talking Heads Experiment*. Antwerp: limited pre-edition for the Laboratorium exhibition

Stefanskaia, I. (2005) *Regelbasierte Wortformerkennung für das Deutsche als Grundlage eines linksassoziativen Sprachmodells für einen automatischen Spracherkenner mit Prototyping einer Sprachsteuerungskomponente zur Steuerung dezidierter Arbeitsabläufe in einem radiologischen Informationssystem*, M.A. thesis, CLUE

Talmy, L. (1985) "Lexicalization patterns: semantic structure in lexical forms," in T. Shopen (ed.), p. 57–149

Tarr, M.J., and H.H. Bülthoff (1995) "Is human object recognition better described by Geon structural descriptions or by Multiple Views? Comments on Biedermann and Gehardstein (1993)," *Journal of Behavioral Psychology: Human Perception and Performance*, Vol. 21(6):1494–1505

Tarski, A. (1935) "Der Wahrheitsbegriff in den Formalisierten Sprachen," *Studia Philosophica*, Vol. I:262–405

TCS = Hausser, R. (1992) "Complexity in left-associative grammar," *Theoretical Computer Science*, Vol. 106.2:283–308

Tesnière, L. (1959) *Éléments de syntaxe structurale*, Paris: Editions Klincksieck

Thalheim, B. (2000) *Entity-Relationship Modeling: Foundations of Database Technology*, Springer

Thiel, C. (1995) *Philosophie und Mathematik*, Darmstadt: Wissenschaftliche Buchgesellschaft

Tilden, M.W., and B. Hasslacher, (1996) "Living machines," Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Tittel, A. (2008) *Optimierung der deutschen JSLIM-Morphologie*, M.A. thesis, CLUE

Tomasello, M. (2003) “Introduction: Some surprises to psychologists,” in M. Tomasello (ed.) *The New Psychology of Language, Vol. II*, Mahaw, N.J.: Lawrence Erlbaum

Tompa, F. (1992) “An Overview of Waterloo’s Database Software for the OED,” technical report: OED-92-01

Twiggs, M. (2005) *Eine Behandlung des Passivs im Rahmen der Datenbanksemantik,* M.A. thesis, CLUE

Vorontsova, L. (2007) *Implementierung eines automatischen Wortformerkennungssystems für das Russische in JSLIM*, M.A. thesis, CLUE

Webb, B., and J. Wessnitzer (2009) ”Perception for action in insects,” in Arena, P., and L. Patanè (eds.)

Weber, C. (2007) *Implementierung eines automatischen Wortformerkennungssystems für das Italienische mit dem Programm JSLIM*, M.A. thesis, CLUE

Weber, C. (2011) *Syntaktisch-semantisches Parsen des Italienischen*, Ph.D. dissertation, CLUE. Frankfurt am Main: Peter Lang Verlag 2012

Weber, C., and J. Handl (2010) “A base-form lexicon of content words for correct word segmentation and syntactic-semantic annotation”, in M. Pinkal, J. Rehbein, S. Schulte im Walde, and A. Storrer (eds.) *Proceedings of the KONVENS 2010*, p. 175–179, Saarbrücken: universaar

Weiss, M.A. (2005) *Data Structures and Problem Solving Using Java. 3rd ed.*, Upper Saddle River, NJ: Pearson Addison-Wesley

Weizenbaum, J. (1965) “ELIZA – A computer program for the study of natural language communications between man and machine,” *Comm. ACM*, Vol. 9.1 (11)

Wetzel, C. (1996) *Erstellung einer Morphologie für Italienisch in Malaga*, CLUE-betreute Studienarbeit der Informatik, Univ. Erlangen–Nürnberg

Whorf, B. (1964) *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf*, J. B. Carroll (ed.), MIT Press

Wiener, N. (1948) *Cybernetics: Or the Control and Communication in the Animal and the Machine*, Cambridge, MA: MIT Press

Wierzbicka, A. (1991) *Cross-Cultural Pragmatics: The Semantics of Human Interaction*. Berlin, New York: Mouton de Gruyter

Wilson, E.O. (1998) *Consilience: The Unity of Knowledge*, New York: Alfred Knopf

Wu, Z., and M. Palmer (1994) "Verb semantics and lexical selection," Proceedings of the 32nd annual meeting of the ACL, p. 133–138

Wunderlich, D. (2006) "Towards a structural typology of verb classes," in D. Wunderlich (ed.) *Advances in the Theory of the Lexicon*, Berlin: Mouton de Gruyter, p. 58–166.

Wunderlich, D. (ed.) (1976) *Wissenschaftstheorie der Linguistik*, Kronberg: Athenäum (= Athenäum Taschenbücher Sprachwissenschaft)

Wurmbrand, S. (2001) *Infinitives. Restructuring and Clause Structure*. Studies in Generative Grammar 55. Berlin: Mouton de Gruyter

Yates, P. (1968) *Bullit*, Hollywood: Warner Brothers/Seven Arts

Yovits, M.C. (ed.) (1992) *Advances in Computer Science, 2nd ed.*, Academic Press

Zheng, W. (2011) *Verbalkomplexe im Chinesischen, eine Implementierung in JSLIM*, M.A. thesis, CLUE

Zholkovskij, A., and Mel'chuk, A. (1965) "O vozmozhnom metode i instrumentax semanticheskogo sinteza." (On a possible method and instruments for semantic synthesis.) Nauchno-texnicheskaja informacija. http://www.neuvel.net/meaningtext.htm

Zipf, G.K. (1932) *Selected Studies of the Principle of Relative Frequency in Language*, Cambridge, MA.: Harvard U. Press

Zipf, G.K. (1935) *The Psycho-Biology of Language*, Cambridge, MA: MIT Press

Zipf, G.K. (1949) *Human Behavior and the Principle of Least Effort*, Cambridge, MA: Addison-Wesley

Zischler, L. (2011) *Implementierung eines Syntaxparsers für das Chinesische*, M.A. thesis, CLUE

Zue, V., R. Cole, and W. Ward (1995) "Speech recognition," in R. Cole (ed.) 1997

Zweig, E. (2008) *Dependent Plurals and Plural Meaning*, Ph.D. dissertation, The University of York

Name Index

Subject Index

Backcover

The construction of a talking robot may be approached from two ends: bottom up and top down. The bottom up approach starts with the hardware and wrestles with mechanical aspects, as when building an artificial centipede. The top down approach, in contrast, is theory-oriented and begins with designing the general component structure of an artificial cognitive agent with language.

This book takes a top down approach. The software for free human-machine communication in a natural language and domain of choice is based on a theory of language called Database Semantics. DBS was developed to answer the central theoretical question for building a talking robot:

How does the mechanism of natural language communication work?

For doing what is requested and reporting what is going on, a talking robot requires not only language but also nonlanguage cognition. The contents of nonlanguage cognition are re-used as the meanings of natural language surfaces.

Agent-externally, DBS handles the language-based transfer of content by using nothing but modality-dependent unanalyzed external surfaces such as sound shapes or dots on paper, produced in the speak mode and recognized in the hear mode. Agent-internally, DBS reconstructs cognition by integrating linguistic notions like functor-argument and coordination, philosophical notions like concept-, pointer-, and baptism-based reference, and notions of computer science like input-output, interface, data structure, algorithm, database schema, and functional flow.

Presented as a declarative specification, the DBS robot is designed as an abstract software machine which is formalized independently of a particular choice of programming language or hardware. When running on today's standard computers, concepts like *take, blue*, or *square* are represented by English words as place holders. Their literal meaning in English speaking humans is used as a temporary substitute for the corresponding recognition and action procedures in an artificial cognitive agent (grounding).