Computational Cognition

Integrated DBS Software Design for Data-Driven Cognitive Processing

December 9, 2019; revised July 5, 2020

ROLAND HAUSSER

Friedrich Alexander Universität Erlangen Nürnberg (em.)

email: rrh@linguistik.uni-erlangen.de

Start May 07, 2017
CC1 July 29, 2017
CC2 March 05, 2018
CC3 June 27, 2018
CC4 August 03, 2018
CC5 September 20, 2018
CC6 October 01, 2018
CC7 November 01, 2018
CC8 December 03, 2018
CC9 January 02, 2019
CC10 February 16, 2019
CC11 March 11, 2019
CC12 April 5, 2019
CC13 May 2, 2019
CC14 June 14, 2019
CC15 July 2, 2019
CC16 December 9, 2019
CC17 July 5, 2020 (main changes in Chaps. 9, 12, 13)

河宇士

Pen name given to the author by Professor Inseok Yang, President of the Korean Society of Linguistics, Seoul 1982.

Preface

It is shown by the history of science that a basic property inadvertently omitted in the beginning can not be added post hoc. Therefore a theory of computational cognition should aim from the outset to be as complete as possible and draw from all three branches of modern science, i.e. the natural sciences, the engineering sciences, and the humanities (grammar, philosophy of language).

The natural and the engineering sciences have long evolved a symbiotic relationship, but the humanities still stand apart. Designing and building a talking robot, however, is a challenge for which all three are needed. Agent-based Database Semantics (DBS) integrates recognition and action interfaces, an onboard orientation system, a content-addressable database, a data structure of nonrecursive feature structures with ordered attributes, and a linear algorithm.

I Model in Agent or Agent in Model?

The empirical scope of a scientific theory is determined by its ontology. The DBS ontology for building a computational model of cognition is agent-based like the natural prototype: the artificial agent looks out into the world and interacts with it autonomously. The world is given in the form of raw data and any interaction is solely by means of (i) the agent's interfaces for recognition and action, and by (ii) cognition-internal reasoning on content stored in memory. In this sense, the agent-based ontology places the model inside the agent.

The alternative ontology is sign-based: an introspector-definer analyzes the intuitive relation between language signs and an abstract universe of discourse, called model. If the model were to include an agent, it would be defined like tables or chairs, i.e. as a virtual, immaterial doll without any interfaces for autonomous recognition and action; what it perceives or does would be entirely a matter of definition in a metalanguage by the introspector-definer. In this sense, agents are inside the models of a sign-based ontology.

An agent-based ontology has a broader empirical base than a sign-based ontology: (i) artificial interfaces for the sensory modalities, (ii) content in memory resonating with current processing, (iii) switching between the speak and the hear mode, and (iv) switching between language and nonlanguage cognition are indispensable in a DBS robot, but abstracted away from in sign-based systems. Conversely, instead of the sign-based definition of artificial models in a metalanguage (if-then conditionals based on truth values), the agent-based ontology is more realistic because it treats the world as given.

II Data-Driven or Substitution-Driven?

The dichotomy between an agent-based and a sign-based ontology is complemented by the dichotomy between a data-driven and a substitution-driven computation. The input to a data-driven system is provided by recognition, memory, or a preceding operation; the output is content for action, provided by current processing or blueprints retrieved from memory.

A substitution-driven system, in contrast, creates a hierarchy with rules which replace an abstract node with a larger (top down expansion) or smaller (bottom up reduction) expression. Building on the work of Frege (1848–1925), substitution-driven formalisms were used by Hilbert (1862–1943), Russell (1872–1970), Tarski (1901–1983), and Gödel (1906–1978) for the development of axiomatic systems and resulted in recursion, automata, and computational complexity theory, such as the rewrite systems of Post (1936).

When Chomsky borrowed the substitution-driven approach for analyzing natural language in the form of "generative grammar," he ran into the problem that the extremely successful original was not designed to provide distinctions between recognition and action, the speak and the hear mode, and language and nonlanguage content. Also, its 'vertical' derivation order is inherently in conflict with the 'horizontal' time-linear structure of natural language, artificially creating the "problem of serialization" for generative grammar. The way out was using the same start symbol S for randomly computing different substitutions to generate "base structures" alleged to be innate and universal.

DBS, in contrast is data-driven. In the hear mode, the ordered input of concrete agent-external surfaces is lexically analyzed as proplets which are connected into content with the classical semantic relations of structure coded by address. Navigating along the semantic relations between the order-free proplets of a content activates the proplets traversed, making them input to the language-dependent surface realization of the speak mode.

Regarding computational complexity, data-driven LA-grammar provides the first, and so far the only, formal language hierarchy (TCS) which is orthogonal to the Chomsky hierarchy of substitution-driven Phrase Structure Grammar. It has been shown (FoCL Sects. 12.5, 22.3) that the natural languages are in the language class of C1-LAGs, which parses in linear time.

III Media and their Dual Modalities

The interaction between the agent and the raw data provided by its environment is based on the *sensory* media. Each medium has two modalities, one for recognition and one for action. For example, the sensory medium of speech has the dual sensory modalities of vocalization for action and audition for recognition, writing has the dual sensory modalities of manipulation for action and vision for recognition, and accordingly for Braille and signing (11.2.1).

In addition to the sensory media and modalities there are the *processing* media and modalities (11.2.2), which deal solely with content (no raw data). Examples are (i) natural cognition based on the electrochemical medium and (ii) artificial cognition based on the medium of a programming language. The dual processing modalities of a DBS robot are (a) the *declarative* (alphanumeric) commands in a programming language, written by software engineers to be interpreted for recognition and action by a computer, and (b) the *procedural* (electronic) modality for executing the declarative commands automatically.

IV Functional Equivalence and Upscaling Continuity

The cognition of an artificial agent must be functionally equivalent to the natural prototype at a certain level of abstraction. For example, if the artificial agent is able to spontaneously call a color or a geometric shape the same as a human, functional equivalence is achieved. Similarly for the compositional aspect: if the artificial agent understands the difference between the dog bit the man and the man bit the dog the same as an English-speaking human, there is functional equivalence. In such instances proper functioning may be verified by tracing the software via the service channel of the artificial agent (1.1.1).

That computational (1) string search, (2) pattern matching, and (3) iteration are essential for building functional equivalence in a robot does not imply that natural cognition must use these methods as well. In analogy, even though a horse and a motorcycle are functionally equivalent at a certain level of abstraction, i.e. for one or two persons getting from A to B, the functioning of a motorcycle has no bearing on the biological analysis of a horse.

Functional equivalence is counterbalanced by a second, complementary standard of adequacy, namely the upscaling continuity of test cycles. A cycle consists of (a) testing the current software version automatically on systematic data (test suite), (b) correcting all errors, and (c) extending the test suite to additional data, after which the next test cycle is started. A cycle is successful if functional equivalence is achieved. Upscaling continuity holds, if the cycle following a successful cycle is successful as well.

V Abstraction in Programming: Declarative Specification

Programming languages are in a continuous process of debugging and development (updates, new releases). This requires software maintenance which may be labor-intensive; for example, a program written in 1985 in Interlisp-D will require work by an expert in order to run in today's Common Lisp. Also, computer programs, old or new, are difficult for humans to read and without additional explanation do not easily reveal the conceptual idea.

Therefore there have long been efforts at defining programs at a higher level of abstraction such as UML (unified modeling language) and ER (entity relationship model). The goal is to use notions which are meaningful to humans, but also translate into various programming languages.

It turns out, however, that the general purpose aspiration of UML, ER, and similar proposals, i.e. working for any task and any programming language, causes massive overhead and may require more work than they can actually save. This holds especially for research outside of today's well-established computational applications based on substitution.

Therefore the data structure, the database schema, and the algorithm of DBS are defined directly but abstractly: they are custom-designed to handle the software tasks inherent in the cognition of a talking robot (15.2.2) in a simple, standardized, conceptually transparent, and empirically comprehensive manner, called *declarative specification*.

VI Symbolic vs. Subsymbolic

The dichotomy between the processing modalities of natural and artificial agents is complemented by the dichotomy between symbolic and subsymbolic processing. The design of cognition in DBS is symbolic because it (a) uses the classical semantic kinds of concepts, indexicals, and names, and the classical semantic relations of functor-argument and coordination, intra- and extrapropositionally, for language and nonlanguage content, and (b) is based on the use of a programming language. This is in contradistinction to connectionism, which is computational but subsymbolic (Miikkulainen 1993).

VII Now Front and the On-Board Orientation system

The arena for processing the artificial agent's current recognition and action is the *now front*, defined as part of the on-board memory (database). On the one hand, data at the now front subactivate resonating content in storage. On the other hand, systematic clearance of the now front limits the data available as input to the data-driven application of DBS operations.

Proplets at the now front are provided (i) by current recognition (monitoring), (ii) memory, and (iii) processing for action. The now front is cleared in regular intervals by moving into fresh memory territory, leaving the proplets which have ceased to be candidates for further processing behind in permanent storage (*loom-like clearance*). This results automatically in a strictly time-linear storage of proplets in the token lines of the artificial agent's A-memory.

Proplets are defined as nonrecursive feature structures with ordered attributes, connected into content by the semantic relations of structure coded by address. In monitoring, each proposition is anchored to a 'STAR,' which stands for Space (location), Time, Agent, and intended Recipient. The values of these attributes are provided by the agent's on-board orientation system; they specify (i) the current content's origin and (ii) provide indexicals with values to point at.

VI Data-Driven Behavior

The computational backbone of DBS cognition is the continuous monitoring of current recognition and action in the agent's ecological niche. Based on computational pattern matching, behavior is data-driven. Its basic options include switching between the speak and hear mode (turn taking), interaction between language and nonlanguage cognition (reference), and reasoning in the think mode based on inferencing (blueprints for nonlanguage action as well as production and interpretation of literal and nonliteral language use).

The agent's cognition has two basic sources of data: (i) current recognition and action, and (ii) resonating content in memory. Resonating content is activated by current content at the now front and may participate in current processing by means of shadowing. Shadowing copies the proplets of a resonating content by address to the now front. This allows content stored in memory to participate in current processing without touching the original data.

The content stream resulting from the agent's interaction with a changing environment is stored in the agent's memory, which is content-addressable and utilizes the contents' time-linear arrival and departure structure in storage and retrieval. The uniform time-linear derivation order throughout DBS cognition supports a basic design feature of DBS, namely the treatment of the narrative speak mode as an optional reflection of a time-linear think mode navigation in the form of accompanying language-dependent surfaces.

In terms of computational complexity theory, a data-driven time-linear derivation may split into several parallel path, whereby the worst case is recursive ambiguity (exponential complexity). However, recursive ambiguity does not occur in the natural languages, which are in the class of C1-LAGs and thus of linear complexity. Needless to say, low computational complexity is important for the real time performance of an artificial cognitive agent.

VIII Structure of the book

The book has three parts, each part has five chapters, and each chapter has six sections. The front matter consists of the front cover, this preface, followed by the table of content, and a list with a selection of preceding works, including their acronyms, e.g. FoCL for *Foundations of Computational Linguistics*, used for brief and easy reference. The back matter consists of the bibliography, a name index, a subject index, and a back cover. Each chapter opens with an abstract and contains copious cross-references to related content in other parts of the book. Online, cross-referencing is automatic (`\hyperref`).[1]

Part I begins with the ontological distinction between (i) a sign-based and (ii) an agent-based approach, and continues with explanations of the data structure; the content-addressable database schema; the time-linear derivations of the speak and the hear mode; resonating content; induction, deduction, and abduction in inferencing; and concludes with a reconstruction of eight classical syllogisms as a test suite for DBS inferencing in the think mode.

Part II complements the literal use of language in the speak and hear mode with a reconstruction of syntactic mood adaptations and figurative use. It concludes with special situations such as overhearing by an unintended hearer, the effect of re-reading a text from science, and a classic example of performative use. The database schema of DBS is shown to lend itself not only to the tasks of traditional storage and retrieval, but also of reference, coreference, shadowing, coactivation of resonating content, and selective activation.

Part III complements the treatment of individual topics in linguistics, analytic philosophy, and cognitive psychology with an overall software structure in the form of three interacting main components, called the interface, the memory, and the production component. The two final chapters summarize basic differences between agent-based DBS and today's sign-based approaches. The back cover lists some of the issues addressed in the form of key words.

[1] A mouse click on a reference, e.g. 1.1.2, brings up the page with the example. Depending on the editor, typing "[," "CTRL [," or clicking the go and then the back button in the control bar at the top of the Adobe pdf window brings back the point of departure. Eventually, given enough interest, FoCL3, NLC2, CLaTR2, TExer, and the present volume should be combined into a single electronic file with automatic cross-referencing working also between volumes.

Contents

Abbreviations Referring to Preceding Work

References to FoCL, NLC, and CLaTR are to the latest editions.

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; preprint 2nd Ed. 2017, pp. 363, 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
CLaTR	Hausser, R. (2011) *Computational Linguistics and Talking Robots, Processing Content in Database Semantics*, pp. 286, Springer; preprint 2nd Ed. 2017, pp. 366, at `lagrammar.net`
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. (2016) *How to build a Talking Robot – Linguistics, Philosophy, and Artificial Intelligence*, pp. 120, Springer, in print

CTGR Hausser, R. (2017) "A computational treatment of generalized reference," *Complex Adaptive Systems Modeling*, Vol. 5(1):1–26, DOI: 10.1186/s40294-016-0042-7. Also available at `http://link.springer.com/article/10.1186/s40294-016-0042-7`. For the original with section and example numbering see `lagrammar.net`

TExer Hausser, R. (2017) *Twentyfour Exercises in Linguistic Analysis – DBS Software Design for the Hear and the Speak Mode of a Talking Robot*, DOI: 10.13140/RG.2.2.36392.01282, `lagrammar.net`

This book:

CC Hausser, R. (2019) *Computational Cognition, Integrated DBS Software Design for Data-Driven Cognitive Processing*, pp. i–xii, 1–237, `lagrammar.net`

Part I

Semantics

1. Ontology

The term *ontology* may be transliterated as ‘account of what there is.’ The ontology of a field of science comprises the basic elements and relations assumed to allow a complete analysis of its phenomena. For example, the Presocratics tried to explain nature based on an ontology of fire, water, air, and earth (Fairbanks 1898). Today, the ontology of physics is based on a space-time continuum, protons, electrons, neutrons, quarks, neutrinos, etc. (Dürr 2015).

Similarly in theories of meaning in philosophy. There was a time in which meaning was based on naming; for example, the celestial body rising in the morning and setting in the evening served as the meaning of the word sun. Then meaning became defined in terms of set-theoretic denotations in possible worlds. Which ontology is required for building a computational cognition?

1.1 Scientific Observation

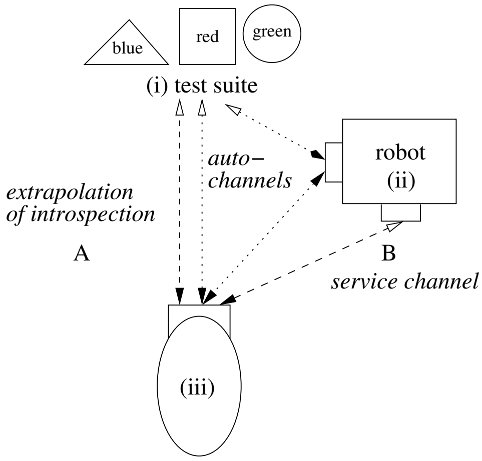
The goal of building a talking robot is functional equivalence between the artificial agent and its natural prototype. The equivalence is defined at a level of abstraction which is above the distinction between different processing media (11.2.1), such as natural, mechanical, and electronic processing. Functional equivalence is shown, for example, by the basic operations of arithmetic: 3+4 equals 7 no matter whether the calculation is performed by (i) a human,[1] (ii) a mechanical calculator, or (iii) a computer.

For establishing functional equivalence between the cognition of a natural and an artificial agent, two basic observation set-ups must be accommodated. In one, the scientist observes the nonlanguage interaction of the artificial agent with its environment (1.1.1). In the other, the scientist observes the language interaction between an artificial and a natural agent (1.1.2). As shown in 1.1.3, the two set-ups may be combined.

[1] The operations of arithmetic as they are processed by the human brain are described by Menon (2011).

As an example of the first constellation, consider the following nonlanguage set-up, which is an 'external' variant[2] of Winograd's (1972) blocks world:

1.1.1 ROBOT'S NONLANGUAGE COGNITION OBSERVED BY SCIENTIST



The test suite, the robot, and the scientist are items in close proximity in an external environment. Their bodies and their interaction can be verified not only by human observers viewing the setup, but also be recorded by machines of the natural and the engineering sciences. Within the setup, there are two kinds of channels, the (i) naive, spontaneous autochannel (indicated as ....) and the (ii) privileged, reasoned science channels A and B (indicated as - - - - -).

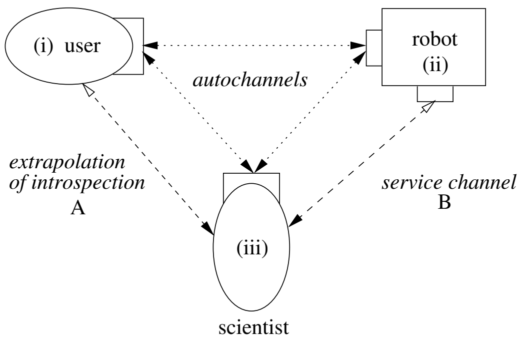
The autochannels connect the test suite, the robot, and the scientist. They allow the scientist to naively compare his or her interaction with the test suite, e.g. putting one object on top of another, with that of the robot. They also allow the robot to naively compare its own test suite interaction with that of the scientist (◄··►).

The privileged science channels A and B, in contrast, connect the scientist with the test suite and with the robot, but without any inverse connections (◄··▻). The extrapolation of introspection channel A complements the scientist's naive autochannel cognition with the introspection of the scientist and the insights of the natural sciences as the basis for constructing the robot's artificial interfaces. The service channel B allows the scientist to determine directly whether or not the robot's software operations work as intended (tracing) and is the basis for debugging in the current upscaling cycle.

[2] For practical reasons, the original blocks world is 'internal' in that it represents recognition and action as simulations on the computer screen. Today, almost fifty years later, an 'external' approach is crucial because it necessitates building the interface component for the autonomous interaction between an external artificial agent and the raw data of its environment.

Just as computational nonlanguage cognition requires a robot and an external environment (test suite) but no partner of discourse (3.1.3, immediate reference), computational language cognition requires a robot and a partner of discourse but no external environment (3.1.2, mediated reference):

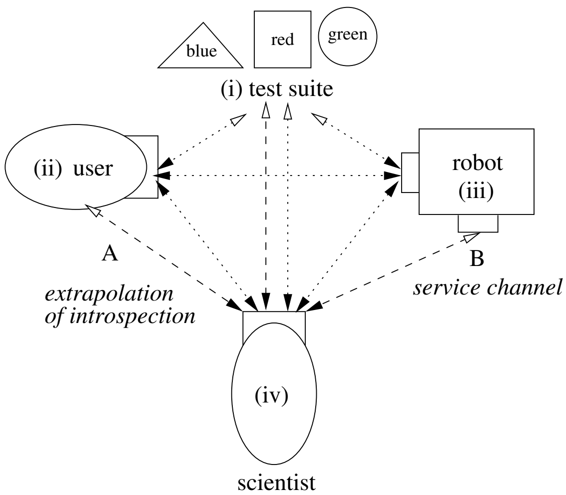
1.1.2 ROBOT'S LANGUAGE COGNITION OBSERVED BY SCIENTIST



The scientist observes the language behavior of (i) the user and (ii) the robot via the autochannels, e.g. the scientist hears and understands what they say and can also interview them about it. In addition, the scientist observes the cognitive states of the user indirectly via a scientifically founded extrapolation of introspection (channel A) and the robot directly via the service channel B. For the scientist, the user and the robot are equally real, and at the level of functional equivalence their cognitive states have the same ontological status.

The constellations 1.1.1 and 1.1.2 combine as follows (3.1.1, immediate reference):

1.1.3 COMBINING LANGUAGE AND NONLANGUAGE OBSERVATION



The autochannel is available to the user, the robot, and the scientist. It is the channel employed most, but also most prone to error. For example, in nonlanguage cognition there are the visual illusions and in language cognition there are the misunderstandings. In addition, there is the possibility that the partner of discourse might not be telling the truth.

1.2 Agent-Based vs. Sign-Based

The ontology inherent in the constellations 1.1.1–1.1.3 is agent-based insofar as it distinguishes between the agent's cognition-external environment (raw data) and the agent-internal cognition which interacts with its environment solely by means of sensors and actuators. This ontology may be summarized as follows:

1.2.1 ONTOLOGY OF AN AGENT-BASED APPROACH

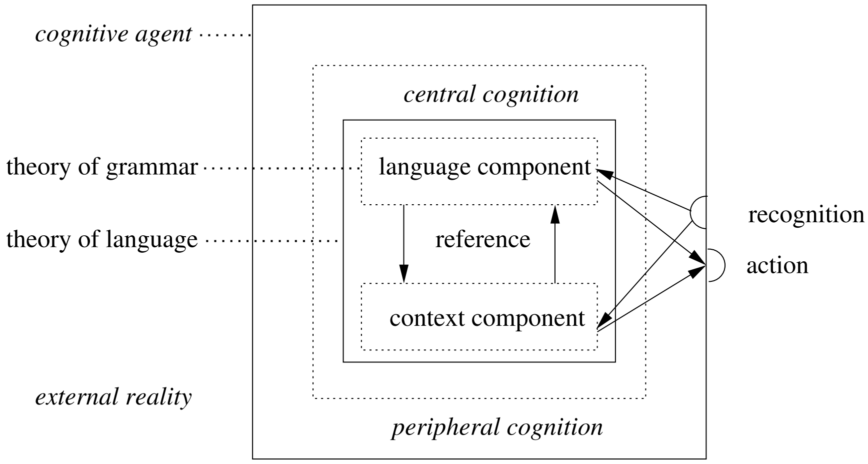
1. **Ecological niche**
   The agent's ecological niche is treated as given.[3] It may be natural or artificial, but it must be real (may not be virtual).
2. **A-memory**
   A-memory (Sect. 2.3) is the section of the agent's on-board database for storing episodic and generic content resulting from connecting concepts represented by place-holder values.
3. **B-Memory**
   B-memory (Sect. 12.1) is the section which provides declarative definitions and procedural implementation for placeholder values of complex concepts like cook (12.1.2).
4. **C-Memory**
   C-memory (Sect. 11.3) is the section which provides declarative definitions and procedural implementations for placeholder values of elementary concepts like blue (11.3.2).
5. **Semantic Relations of Structure**
   The concepts in the agent's A-memory are connected into complex content[4] by means of the classical semantic relations of structure, i.e. functor-argument and coordination, intra- and extrapropositionally (Sect. 2.5).

[3] Brooks (1986): “The world is its own best model.”
[4] Language contents are called signs. Those represented by concepts are called *symbols* by Peirce.

6. **Speak Mode**
   The speak mode maps an agent-internal content into raw data, i.e. agent-external language-dependent surfaces in a sensory modality (11.5.2).
7. **Hear Mode**
   The hear mode maps raw data, i.e. agent-external language-dependent surfaces in a sensory modality, into an agent-internal content (11.4.1, 11.4.3).

The structure of a cognitive agent capable of communicating with natural language may be shown conceptually as follows:

1.2.2 PRELIMINARY STRUCTURE OF AN AGENT WITH LANGUAGE



Cognition is shown as consisting of two separate but interacting components, the language and the context component. The language component is subject of the theory of grammar. The interaction between the language and the context component is subject of the theory of language.[5] The language and the context component use the same interfaces for recognition and action.

As a preliminary[6] conceptual view, 1.2.2 characterizes the most basic mechanisms of natural language communication. In the speak mode, content at the context level is mapped into content at the language level (↑) and realized as unanalyzed, modality-specific, language-dependent, external surfaces by

[5] For a classic comparison of Theory of Grammar and Theory of Language see Lieb (1976).

[6] It became apparent that separate, self-contained language and context components are unsuitable for modeling their cognitive interaction and are therefore unsuitable for programming (FoCL Sect. 22.2). The computational solution in DBS is placing the matching between elementary language and nonlanguage contents (proplets) into the structured A-memory of DBS (6.5.1).

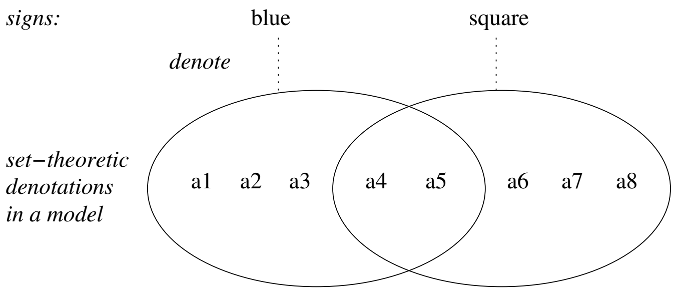
the agent's action component (↘). In the hear mode, unanalyzed modality-specific external surfaces (↖) are interpreted by the agent's recognition component as language content and mapped into the context component (↓).

The basic alternative to the agent-based ontology is the sign-based ontology. It has a continuous tradition from the Presocratics (Heidel 1903) to today, either implicitly (Predicate Calculus) or explicitly (Sag 2012). Taking the perspective of an introspector-definer,[7] the sign-based ontology bypasses the cognitive agent by defining meaning as a direct relation between the sign and (a model of) the world (1.2.3). There is no room for an agent interacting with its environment (as in 1.1.1, 3.1.3), for agents interacting with each other (as in 1.1.2, 3.1.1, 3.1.2), a speak and a hear mode, an on-board orientation system, or a memory.

Symbolic logic, for example, defines contingent (nonlogical) meaning as a direct relation between language and a set-theoretic universe of discourse called *model* or *model structure*. Intended as an abstraction of the world, the term *model* is used if a single state is defined, formally the triple <A, B, F>, where A is a set of items, B a set of basic expressions (signs, surfaces, content words), and F a denotation function from B into the power-set over A, written $\mathscr{P}(A)$. The empty set $\emptyset$ and the set $\{\emptyset\}$ containing the empty set are by definition elements of $\mathscr{P}(A)$ and utilized as the values true and false, respectively. A model which is extended by adding the sets I of possible worlds and J of possible moments of time for treating logical modality is called a *model structure*, formally <A, I, J, B, F>, where A, I, and J are infinite.[8]

For example, the sign-based approach defines the meaning of the word blue as denoting the set of all blue things and the word square as denoting the set of all squares in a model. Their combination, i.e. blue square, denotes the intersection of blue and square and may be shown graphically as follows:

1.2.3 Defining contingent meaning in set theory



[7] See Barwise and Perry (1983), p. 226.

[8] In Montague's intensional logic, the dependence of a proposition like [walk'(b)] for Bill walks on the model structure @ and parameter values in A, I, and J is shown by adding a superscript, as

This intuitive sketch is formalized by defining the denotation function F (Montague 1973, PTQ)[9] for blue and square as follows:

1.2.4 Two denotation functions

F(blue) = {a1, a2, a3, a4, a5}
F(square) = {a4, a5, a6, a7, a8}
where the signs blue and square ε B and a1–a8 ε A of the model <A,B,F>.

In this way, the aspects of an agent's interaction with the real world are abstracted away from. Whether a1, for example, happens to be blue, or red, or a square, or a unicorn is entirely a matter of arbitrary definition.[10] Relative to the model 1.2.3, the proposition $[\text{blue}(x) \wedge \text{square}(x)]^{A,I,J,g}$ is true for arbitrarily setting the variable assignment g(x) (NLC Sect. 6.4) to a4 or a5 in accordance with 1.2.4 (intersection), and false otherwise.

A system's definition specifies its ontology. As an example consider the following specification of first order Predicate Calculus, freely borrowing notions from Montague, but skipping typed λ-calculus, the separation of syntax and semantics, and the distinction between intension and extension (Sect. 15.3).

1.2.5 Ontology of sign-based Predicate Calculus

1. **Model <A, B, F>**,
   where A is an infinite set of entities (PTQ p. 257), B a finite set of basic expressions (PTQ p. 250), and F a denotation function with B as the finite domain and $\mathscr{P}(A)$ as the infinite range (PTQ p. 258).[11]
2. **Truth Values 0 and 1**,
   where 0 is defined as $\emptyset$ and 1 as $\{\emptyset\}$ in $\mathscr{P}(A)$.
3. **Operators**
   where $\forall$ (universal) and $\exists$ (existential) are quantifiers[12], $\wedge$ (and), $\vee$ (or), and $\rightarrow$ (implies) are binary operators, and $\neg$ (negation) is a unary operator.
4. **Variables x, y, z**
   which may be bound to elements of A (items).

---

in $[\text{walk'}(b)]^{@,a,i,j,g'}$. Introducing additional parameters for 1st, 2nd, and 3rd person, as has been suggested, was made light of by Cresswell's (1972, p. 4) joking proposal of adding a "next drink parameter."

[9] After Montague's untimely death in 1971, Prof. Barbara Partee at UCLA almost single-handedly made Montague Grammar the champion of formal semantics.

[10] This arbitrariness is sometimes proffered as "generality," though it leaves a robot deaf and blind.

[11] The page numbers refer to the reprint of PTQ in Montague's collected papers (Thomason ed. 1974).

[12] For a survey of generalized quantifiers in language and logic see Peters and Westersthål (2006).

5. **Recursive Rules**
   which combine well-formed expressions recursively, thereby building up a bracketing structure (Sect. 15.3; PTQ p.258–259).

Just as an ontology without subatomic particles is unsuitable for modern physics, an ontology of computational cognition without an agent, without a distinction between an agent-external reality and agent-internal processing, without interfaces for recognition and action, without a distinction between the speak and the hear mode, and without an agent-internal memory is unsuitable for the task of building a talking robot. A talking robot is not only of interest for a wide range of practical applications, but constitutes the ultimate standard for evaluating the many competing theories of natural language in today's linguistics, language philosophy, language psychology, and computer science.

1.3 Grounding

The semantics of DBS is grounded (Barsalou et al. 2003, Steels 2008, Spranger et al. 2010) and based on the type/token distinction from philosophy. In recognition, concept types (supplied by the agent's memory) are matched with raw data (provided by sensors). In action, a type is adapted to a token for the purpose at hand and realized by actuators as raw data.

A type defines the *necessary* properties of a concept, while a token is an instantiation of the type with additional *accidental*[13] properties (Peirce 1906, CP Vol.4, p. 375). Consider the following example (FoCL Sect. 3.3):

1.3.1 TYPE AND TOKEN OF THE CONCEPT *square*

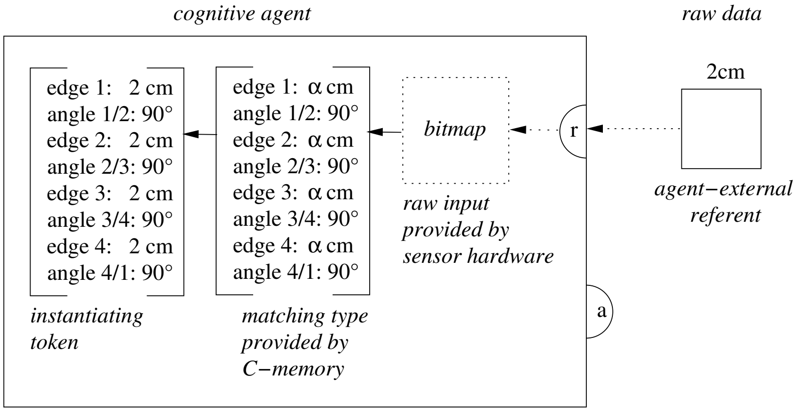
type	*token*
edge 1: α cm	edge 1: 2 cm
angle 1/2: 90°	angle 1/2: 90°
edge 2: α cm	edge 2: 2 cm
angle 2/3: 90°	angle 2/3: 90°
edge 3: α cm	edge 3: 2 cm
angle 3/4: 90°	angle 3/4: 90°
edge 4: α cm	edge 4: 2 cm
angle 4/1: 90°	angle 4/1: 90°
where α is a length	

The type and the token share an ordered column of attributes which specify (i) the number of equally long edges and (ii) the angle of their intersections (necessary properties). The type and the token differ solely in their edge values.

[13] The term accidental is used here in the philosophical tradition going back to Aristotle, 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.

The edge value in the token is the constant 2cm, which is accidental in that the variable α of the type matches an infinite number of square tokens with different edge lengths.

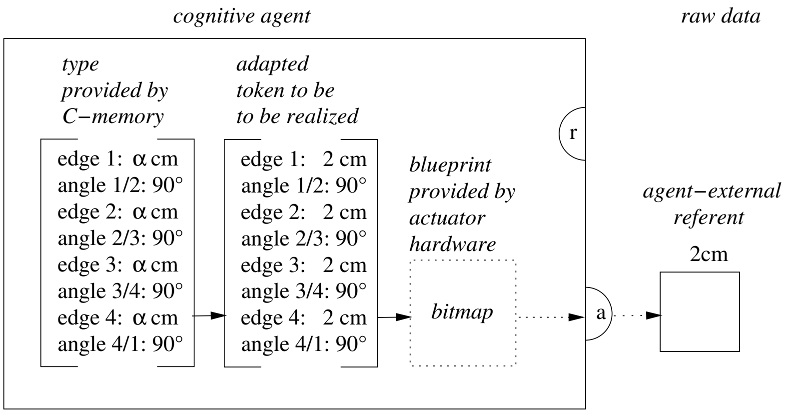
1.3.2 RECOGNITION OF square



The functional order is from right to left: the raw data are supplied by a sensor, here for vision, as input to the interface component. The raw data are matched by the type, resulting in a token. The tokens are used to interpret square place holder values of proplets in A-memory (1.6.2).

The type of square may also be used in action, as when drawing a square:

1.3.3 ACTION OF REALIZING square



The functional order is from left to right: the type is adapted into a token for the agent's current purpose. The token is used as a blueprint for action, which is performed by the agent's interface component in the modality of manipulation (e.g. drawing a square). The general mechanism illustrated in 1.3.2 and 1.3.3 works for all geometric objects for which the sensor and actuator hardware provides sufficiently differentiated raw data and for which the declarative definition and the procedural implementation of the type have been established.

Next consider the recognition and realization of a color, here blue:

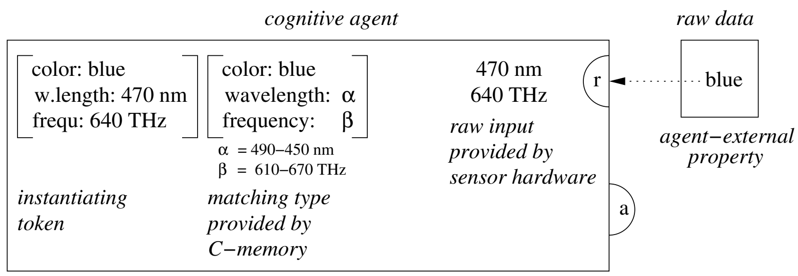
1.3.4 TYPE AND TOKEN OF THE COLOR CALLED blue

type	*token*
$\begin{bmatrix}\text{wavelength: } \alpha \\ \text{frequency: } \beta\end{bmatrix}$ where α ε 490–450 nm and β ε 610–670 THz	$\begin{bmatrix}\text{wavelength: 470 nm} \\ \text{frequency: 640 THz}\end{bmatrix}$

The type specifies the wavelength and the frequency of the color *blue* by means of variables which are restricted to the corresponding intervals provided by physics. The token uses constants which lie within these intervals.[14]

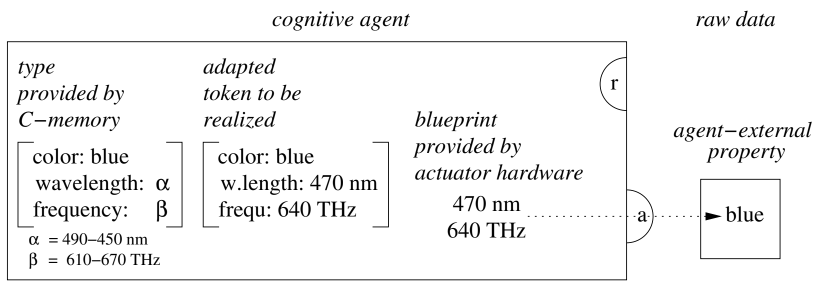
Matching the type with raw data results in an instantiating token (11.3.2):

1.3.5 RECOGNITION OF blue



An example of the corresponding action is turning on the color blue, as a cuttlefish using its chromatophores (11.3.4):

1.3.6 ACTION OF REALIZING blue



The concept type matches different shades of blue, whereby the variables α and β are instantiated as constants in the resulting token. Recognizing the color blue is a general mechanism which may be applied to all colors (11.3.5). It may be expanded to infrared and ultraviolet, and to varying intensity.[15]

Pattern matching based on the type-token relation applies to nonlanguage items (e.g. 1.3.2, 1.3.3, 1.3.5, 1.3.6) and language surfaces (e.g. 11.4.3, 11.5.2) alike. For example, in the surfaces of spoken language the type generalizes over different pitch, timbre, dialect, and speaker-dependent pronunciation. In written language, the type generalizes over the size, color, and font of the letters. Computational type-token matching is more adequate descriptively than the nonbivalent (Rescher 1969; FoCL Chap. 20.5) and fuzzy (Zadeh 1965) logics because type-token matching treats the phenomenon of vagueness at the root (best candidate principle in pattern matching, Sect. 9.2; FoCL Sect. 5.2) instead of tinkering with the truth tables of Propositional Calculus.

1.4 Data Structure

The notion of a data structure is from computer science. The data structure of DBS is a (i) nonrecursive (flat) feature structure with (ii) ordered attributes called *proplet*.[16] Proplets are connected into content by the semantic relations of structure, encoded by writing the address of the 'range proplet' as a continuation value into the 'domain proplet.' This makes semantically connected proplets order-free (CLaTR 3.2.8), which is essential for accommodating their storage and retrieval in the content-addressable database schema (2.3.2, 2.3.3) of the DBS A-memory.

Establishing a bidirectional semantic relation may be shown as a cross-copying of addresses between two proplets. Consider the following concatenation of the adnominal blue as the modifier and the noun square as the modified:

[14] Complementary approaches from cognitive psychology are prototype theory (Rosch 1975) and composition based on geons (Biederman 1987).

[15] Recognition and realization of a concept like blue by an artificial agent is a challenge which can be met with help from the natural and the engineering sciences. The alternative offered by model theory is the arbitrary stipulation of a denotation function like F(blue) in 1.2.4, which by its very nature (ontology) cannot be extended into processing by an agent. This shows how a certain approach, here a sign-based semantics, can not be extended post hoc to a new task. Also, treating the modifier|modified relation blue square as a set intersection (1.2.3) is semantically misguided (15.3.3).

[16] The feature structure of proplets is the direct opposite to the feature structures used in GPSG, LFG, and HPSG (15.4.1), which are recursive with unordered attributes (Carpenter 1992).

As pointed out by Prof. H.J. Schneider (Friedrich Alexander University, Computer Science Department, personal communication), proplets look at first glance like the LISP association list, except for the ordering of attributes. The ordering is similar to the 'records' of Pascal and the 'structures' of Algol 68, C, etc. These, however, may be recursive.

1.4.1 BUILDING A FUNCTOR-ARGUMENT BY CROSS-COPYING

sur:	sur:		sur:	sur:
adj: blue	noun: square		adj: blue	noun: square
cat: adnv	cat: snp		cat: adnv	cat: snp
sem: pad	sem: indef sg	$\Longrightarrow$	sem: pad	sem: indef sg
mdd:	fnc:		mdd: square	fnc:
mdr:	mdr:		mdr:	mdr: blue
nc:	nc:		nc:	nc:
pc:	pc:		pc:	pc:
prn: 17	prn:		prn: 17	prn: 17

The semantic relation is adnominal|noun. It is one of the four functor-argument relations in natural language, the other three being subject/predicate, object\predicate, and adverbial|verb (Sect. 2.5), at the elementary, phrasal, and clausal level of grammatical complexity. The copying in 1.4.1 is from the adj slot of *blue* into the mdr (modifier) slot of *square* and from the noun slot of *square* into the mdd (modified) slot of *blue*.

In a proplet, the second attribute from the top is called the core attribute. For example, in the *blue* and *square* proplets, the core attributes are adj and noun, respectively. The value of a core attribute may be either a language-independent concept, e.g. square, or an indexical, e.g. pro1. A-memory uses English base forms, here blue and square, as placeholders. Their explicit declarative definition and operational implementation is provided by B- (12.1.2) and C-memory (11.3.2, 11.3.4).

The other semantic relation of structure in natural language besides functor-argument is coordination, i.e. conjunction[17] and disjunction. While functor-argument relations cross-copy between different proplet structures (e.g. an adj and a noun in 1.4.1), coordinations cross-copy between proplets of a similar structure. Consider the conjunction of two adnominals:

1.4.2 BUILDING A COORDINATION BY CROSS-COPYING

sur:	sur:	sur:
adj: big	adj: blue	noun: square
cat: adnv	cat: adnv	cat: snp
sem: pad	sem: pad	sem: indef sg
mdd: square	mdd:	fnc:
mdr:	mdr:	mdr: big
nc: blue	nc:	nc:
pc:	pc: big	pc:
prn: 17	prn: 17	prn: 17

[17] Depending on the verb, two kinds of conjunction must be distinguished. For example, John mixed the eggs and the flower does not imply that John mixed the eggs or that John mixed the flower, only that they were mixed together. Mary danced the minuet and the polka, in contrast, implies that Mary danced a minuet and that Mary danced a polka, i.e. separately. Lakoff and Peters (1969) call the former *phrasal* and the latter *sentential* coordination, depending on the underlying "deep structures" assumed. Kempson and Cormack (1981) call the former *collective* and the latter *distributive*.

The cross-copying establishing the functor-argument and coordination relations is from the core value of one proplet to a continuation value of another (2.1.3). Core and continuation features are distinguished by their attributes: noun is a core attribute, while fnc is its continuation attribute, verb is a core attribute, while arg is its continuation attribute, and adj is a core attribute, while mdd is its continuation attribute (NLC A.3.1). In addition, noun, verb, and adj proplets have the continuation attribute mdr for optional modifiers. The operations performing the cross-copying are based on pattern matching (1.6.1).

1.5 Cognitive Foundation

The order, kind, and instantiation(s) of the nine attributes in a standard[18] proplet are as follows:

1.5.1 KINDS AND ORDER OF ATTRIBUTES IN A PROPLET

attributes	*instantiation(s)*		
surface	sur:		
core attributes	noun:	adj:	verb:
category	cat:		
semantics	sem:		
continuation attributes	fnc:	mdd:	arg:
modifier	mdr:		
next conjunct	nc:		
previous conjunct	pc:		
proposition number	prn:		

The core and the continuation attributes each have three instantiations, the other attributes have one. The attributes are used in nonlanguage and language proplets alike. In word form recognition, the proplet structure is also used in function words (2.1.3, line 2), which will be absorbed (2.1.3, line 6).

The values of the sur attributes are the word form surfaces of the natural language at hand. Constant values of the core and continuation attributes are concepts or indexicals (1.5.4). The nc and pc values are the addresses of conjuncts. The prn value is a number.

The terms noun, verb, and adj used as the core attributes of proplets are from linguistics (philology). There are, however, corresponding notions in symbolic logic, namely argument, functor, and modifier, and in philosophy, namely object, relation, and property (FoCL 3.4.1):

[18] Exceptions are (i) the STAR-proplets, which have the six attributes Space, Time, Agent, Recipient, 3rd, and prn (6.3.4, 11.6.1) and (ii) the verb proplets of subordinate clauses, which have an additional fnc attribute for connecting to the higher verb, e.g. the *fall* proplet in 7.6.2, the *change* proplet in 10.2.3, and the *find* proplet in 13.2.1. For a systematic treatment see TExer Sects. 2.5, 2.6, 3.3–3.5.

1.5.2 RELATED NOTIONS IN LINGUISTICS, LOGIC, AND PHILOSOPHY

	(a) *linguistics*	(b) *logic*	(c) *philosophy*
1.	noun	argument	referent (object)
2.	verb	functor	relation
3.	adj	modifier	property

We take it that these are not merely different terms for the same things, but different terms for different aspects of the same things. In particular, the linguistic terminology may be viewed as representing the syntactic aspect, the philosophical terminology as representing the associated semantic aspect and the logical terminology as a preparatory step towards a computational implementation.

In DBS, the distinctions are related as follows:

1.5.3 1ST CORRELATION: SYNTACTIC AND SEMANTIC KIND

	Semantic kind	*Syntactic kind*
1.	referent	noun
2.	property	adn, adv, adnv, intransitive verb
3.	relation	transitive verb

Of the Semantic kinds, we prefer the term referent over object (pace Quine 1960) in order to reserve the latter for the grammatical role (Primus 2012) as in the distinction between subject and object. The term property is interpreted as a 1-place functor and includes (a) adn (adnominal), (b) adv (adverbial), (c) adnv (adnominal or adverbial) modifiers, and (d) intransitive verbs. The term relation is interpreted as a 2- or 3-place functor and includes transitive verbs.

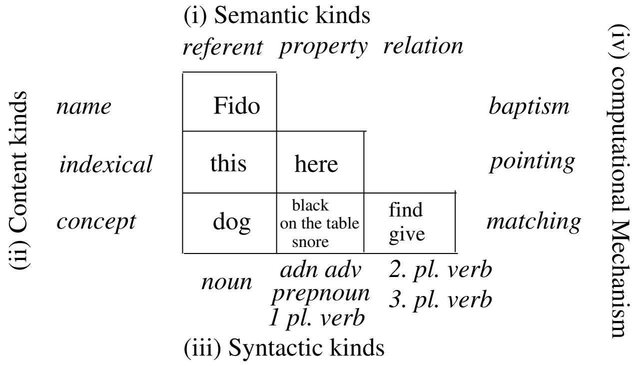
The distinction between (i) Semantic and (ii) Syntactic kinds is complemented by a second, orthogonal pair of triple distinctions, namely three (iii) Content kinds and three associated (iv) Computational mechanisms[19]:

1.5.4 2ND CORRELATION: CONTENT KIND AND COMPUT. MECHANISM

	Content kind	*Computational mechanism*
a.	name	baptism
b.	indexical	pointing
c.	concept	matching

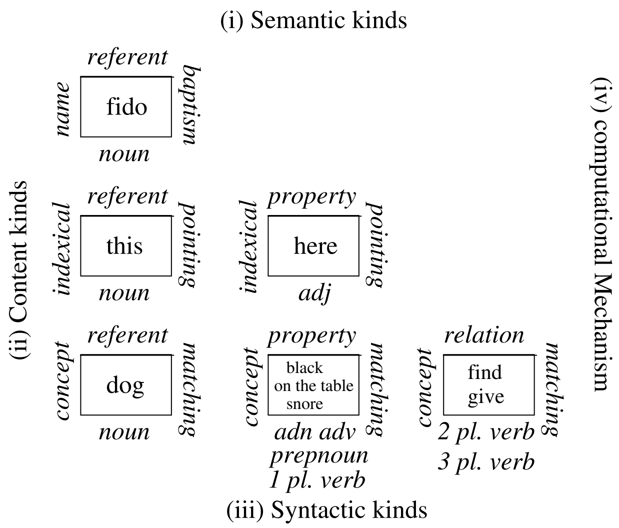
The dichotomies 1.5.3 and 1.5.4 provide 12 ($2{\times}2{\times}3$) basic notions. Empirically, they combine into six classes of proplets which constitute the semantic building blocks of cognition in general and natural language communication in particular. The six classes form what we call the *cognitive square*[20] of DBS:

1.5.5 Cognitive Square of DBS



The twelve basic notions of this NLC 2.6.9 extension are distributed over six basic proplets kinds such that no two are characterized the same:

1.5.6 Closer view of the Cognitive Square



The surfaces inside the rectangles have the following proplet definitions:

---

[19] The terms *matching, pointing* and *baptism* have had informal use in the literature, but without an agent-based ontology. For example, the matching for concepts was used without the type-token relation and its computational implementation based on content and pattern proplets, the pointing for indexicals without an on-board orientation system (STAR), and (iii) baptism without the named referent as the core value for use in the speak and the hear mode.

[20] "Triangle" would be appropriate as well, but the term is used already by cognitive behavioral therapists (CBT). It was also used in the "Semiotic Triangle" by Ogden&Richards' (1923). The term 'square' is well suited to express the orthogonal relation between the Syntactic_kinds/Semantic_kinds and the Content_kinds/Computational_mechanisms.

1.5.7 PROPLETS INSTANTIATING THE COGNITIVE SQUARE OF DBS

$$\begin{array}{lcl}
 & \textit{referent} & \\
\textit{name} & \left[\begin{array}{l}\text{sur: Fido}\\ \text{noun: (dog x)}\\ \text{cat: snp}\\ \text{sem: nm m sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \textit{baptism} \\
 & \textit{noun} &
\end{array}$$

$$\begin{array}{lccl}
 & \textit{referent} & \textit{property} & \\
\textit{indexical} & \left[\begin{array}{l}\text{sur:}\\ \text{noun: this}\\ \text{cat: snp}\\ \text{sem: -hum sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \left[\begin{array}{l}\text{sur:}\\ \text{adj: here}\\ \text{cat: adnv}\\ \text{sem: loc}\\ \text{fnc:}\\ \text{mod:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \textit{pointing} \\
 & \textit{noun} & \textit{adj} &
\end{array}$$

$$\begin{array}{lcccccccl}
 & \textit{referent} & & \textit{property} & & & \textit{relation} & & \\
\textit{concept} & \left[\begin{array}{l}\text{sur:}\\ \text{noun: dog}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \left[\begin{array}{l}\text{sur:}\\ \text{adj: lucky}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{fnc:}\\ \text{mod:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \left[\begin{array}{l}\text{sur:}\\ \text{noun: table}\\ \text{cat: adnv snp}\\ \text{sem: } \textit{on} \text{ def sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \left[\begin{array}{l}\text{sur:}\\ \text{verb: snore}\\ \text{cat: n-s3}' \text{ v}\\ \text{sem: pres}\\ \text{arg:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \left[\begin{array}{l}\text{sur:}\\ \text{verb: find}\\ \text{cat: n-s3}' \text{ a}' \text{ v}\\ \text{sem: pres}\\ \text{arg:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \left[\begin{array}{l}\text{sur:}\\ \text{verb: give}\\ \text{cat: n-s3}' \text{ d}' \text{ a}' \text{ v}\\ \text{sem: pres}\\ \text{arg:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] & \textit{matching} \\
 & \textit{noun} & \textit{adn} & \textit{adnv} & \begin{array}{c}\textit{verb}\\ \text{1-pl}\end{array} & \begin{array}{c}\textit{verb}\\ \text{2-pl}\end{array} & \begin{array}{c}\textit{verb}\\ \text{3-pl}\end{array} &
\end{array}$$

In a proplet, the Semantic kind *referent* is limited to the core attribute noun, *property* is limited to the cat values adn, adv, adnv and to verbs characterized by their cat value as intransitive, and *relation* is limited to verbs characterized by their cat value as transitive.

The Content kind *name, indexical*, and *concept* is specified by a proplet's core value. The corresponding computational Mechanisms *baptism, pointing*, and *matching* are implemented by inserting a 'named referent' as core value into *names*, by indexicals pointing at a STAR value of the onboard orientation system, and by computational type-token matching in the case of *concepts*.

The cognitive square of DBS is empirically important because (i) figurative use is restricted to concepts, i.e. the bottom row in 1.5.5–1.5.7, and (ii) reference is restricted to nouns, i.e. the left-most column. Thus only concept nouns may be used both figuratively and as referents, while indexical properties like here and now may not be used as either, and names only as referents.

1.6 Computational Pattern Matching

The Computational mechanism of pattern matching (1.5.5, iv) is ubiquitous in DBS. Based on the data structure of proplets, i.e. nonrecursive feature structures with ordered attributes, pattern matching is defined by the following constraint (NLC 3.2.3):

1.6.1 DBS MATCHING CONSTRAINT

1. Attribute condition
   The attributes of the pattern proplet must be a sublist (equal or less) of the attributes of a matching content proplet.
2. Value condition
   Each value of the pattern proplet must be compatible with the corresponding value of the content proplet to be matched.

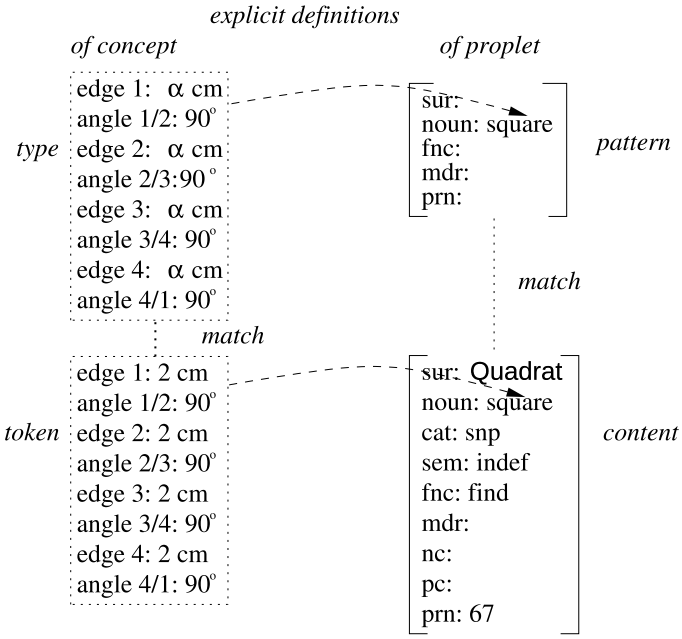
There are two kinds of matching: (i) the type-token relation between concepts and (ii) the relation between variables and constants. Both are based on vertically binding constants to compatible variables:

1.6.2 TWO KINDS OF COMPUTATIONAL PATTERN MATCHING IN DBS

$$
\begin{array}{ccc}
\textit{concept type} & & \textit{pattern proplet} \\
\begin{bmatrix}
\text{edge 1: } \alpha \text{ cm} \\
\text{angle 1/2: } 90^{o} \\
\text{edge 2: } \alpha \text{ cm} \\
\text{angle 2/3: } 90^{o} \\
\text{edge 3: } \alpha \text{ cm} \\
\text{angle 3/4: } 90^{o} \\
\text{edge 4: } \alpha \text{ cm} \\
\text{angle 4/1: } 90^{o}
\end{bmatrix}
& &
\begin{bmatrix}
\text{sur: } \\
\text{noun: } \alpha \\
\text{cat: snp} \\
\text{indef sg} \\
\text{fnc: } \beta \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: K}
\end{bmatrix} \\
\text{*******************}\textit{MATCHING FRONTIER}\text{******************} & & \\
\textit{concept token} & & \textit{content proplet} \\
\begin{bmatrix}
\text{edge 1: 2 cm} \\
\text{angle 1/2: } 90^{o} \\
\text{edge 2: 2 cm} \\
\text{angle 2/3: } 90^{o} \\
\text{edge 3: 2 cm} \\
\text{angle 3/4: } 90^{o} \\
\text{edge 4: 2 cm} \\
\text{angle 4/1: } 90^{o}
\end{bmatrix}
& &
\begin{bmatrix}
\text{sur: Quadrat} \\
\text{noun: square} \\
\text{cat: snp} \\
\text{indef sg} \\
\text{fnc: find} \\
\text{mdr: blue} \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 63}
\end{bmatrix}
\end{array}
$$

The type-token and the variable-constant matching co-occur when the placeholder for a concept is replaced by its explicit definition, as in the following example:

1.6.3 Type-token matching between core values



The matching between the pattern and the content proplet on the right is based in part on the type/token matching of the square concept shown on the left.

A feature structure taking a feature structure as a value, here the feature structure of a proplet taking the feature structure of an explicit core value definition, may raise the question of whether or not the proplets on the right are recursive feature structures after all. The answer is ‘no’ because here the replacement is what is called *static* in computer science: once the placeholder has been replaced by its explicit definition, there is no possibility of another replacement because there is no value in the explicit definition which would support it.

Running the processing of proplets via the semi-universal core values instead of the language-dependent surfaces enables the computational treatment of cognition regardless of the presence or absence of language-dependent sur values (Sect. 3.1). In the debate between universalists and relativists in linguistics, DBS takes the position of a moderate relativism (Nichols 1992; NLC, Sect. 4.6; CLaTR Sect. 3.6).

2. Algorithm

An agent-based approach to building a talking robot requires the reconstruction of (i) nonlanguage recognition, (ii) nonlanguage action, (iii) language interpretation, and (iv) language production. In language production, the speaker realizes content as semantically connected word forms in a sequence of external surfaces (raw data). In interpretation, the hearer interprets the incoming surfaces by lexical lookup and by reconstructing the speaker's semantic relations.

To enable the interaction between language and nonlanguage content in reference, DBS reuses the semantic relations (2.5.1, 2.5.2) of (iii) language interpretation and (iv) production for (i) nonlanguage recognition and (ii) action. To obtain grounding of the language semantics, DBS reuses the concepts of (i) nonlanguage recognition (11.3.2) and (ii) action (11.3.4) for (iii) language interpretation and (iv) production. Without these agent-based design decisions, the computational cognition of DBS could not have been built.

2.1 Time-Linear Hear Mode Derivation

For software development, automatic word form recognition may be run in isolation by using a test list as input. The result is a list of lexical proplets:

2.1.1 ISOLATED LEXICAL ANALYSIS OF Lucy found a big blue square .

sur: Lucy	sur: found	sur: a	sur: big	sur: blue	sur: square	sur: .
noun: [person x]	verb: find	noun: n_1	adj: big	adj: blue	noun: square	verb: v_1
cat: snp	cat: n′ a′ v	cat: sn′ snp	cat: adn	cat: adnv	cat: sn	cat: v′ decl
sem: nm f	sem: past ind	sem: indef sg	sem: pad	sem: pad	sem: sg	sem:
fnc:	arg:	fnc:	mdd:	mdd:	fnc:	arg:
mdr:	mdr:	mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:	pc:	pc:
prn:	prn:	prn:	prn:	prn:	prn:	prn:

For building complex content, a next word proplet (lexical) is connected to the current sentence start by one of the following kinds of hear mode operations:

2.1.2 THE THREE KINDS OF DBS HEAR MODE OPERATIONS

1. cross-copying (connective ×, as in SBJ×PRD; TExer 6.3.1, 1)
2. absorption (connective ∪, as in DET∪CN; TExer 6.3.1, 51)
3. suspension (connective ∼, as in ADV∼NOM; TExer 6.3.1, 32)

A hear mode operation takes two proplets as input and produces one or two proplets as output, as shown by the following derivation:

2.1.3 TIME-LINEAR SURFACE-COMPOSITIONAL HEAR MODE DERIVATION

Lucy	found	a	big	blue	square	•

automatic word form recognition

sur: Lucy	sur: found	sur: a	sur: big	sur: blue	sur: square	sur: •
noun: (person x)	verb: find	noun: n_1	adj: big	adj: blue	noun: square	verb: v_1
cat: snp	cat: n' a' v	cat: sn' snp	cat: adnv	cat: adnv	cat: sn	cat: v' decl
sem: nm f	sem: past	sem: indef sg	sem: pad	sem: pad	sem: sg	sem:
fnc:	arg:	fnc:	mdd:	mdd:	fnc:	arg:
mdr:	mdr:	mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:	pc:	pc:
prn:	prn:	prn:	prn:	prn:	prn:	prn:

syntactic−semantic parsing

	sur: Lucy	sur: found	
1	noun: (person x)	verb: find	*cross−copying*
	cat: snp	cat: n' a' v	
	sem: nm f	sem: past	
	fnc:	arg:	
	mdr:	mdr:	
	nc:	nc:	
	pc:	pc:	
	prn: 14	prn:	

	sur: lucy	sur:	sur: a	
2	noun: (person x)	verb: find	noun: n_1	*cross−copying*
	cat: snp	cat: #n' a' v	cat: sn' snp	
	sem: nm f	sem: past ind	sem: indef sg	
	fnc: find	arg: (person x)	fnc:	
	mdr:	mdr:	mdr:	
	nc:	nc:	nc:	
	pc:	pc:	pc:	
	prn: 14	prn: 14	prn:	

	sur: lucy	sur:	sur:	sur: big	
3	noun: (person x)	verb: find	noun: n_1	adj: big	*cross−copying*
	cat: snp	cat: #n' #a' v	cat: sn' snp	cat: adnv	
	sem: nm f	sem: past ind	sem: indef sg	sem: pad	
	fnc: find	arg: (p. x) n_1	fnc: find	mdd:	
	mdr:	mdr:	mdr:	mdr:	
	nc:	nc:	nc:	nc:	
	pc:	pc:	pc:	pc:	
	prn: 14	prn: 14	prn: 14	prn:	

	sur: lucy	sur:	sur:	sur:	sur: blue	
4	noun: (person x)	verb: find	noun: n_1	adj: big	adj: blue	*cross−copying*
	cat: snp	cat: #n' #a' v	cat: sn' snp	cat: adnv	cat: sn	
	sem: nm f	sem: past ind	sem: indef sg	sem: pad	sem: sg	
	fnc: find	arg: (p. x) n_1	fnc: find	mdd: n_1	mdd:	
	mdr:	mdr:	mdr: big	mdr:	mdr:	
	nc:	nc:	nc:	nc:	nc:	
	pc:	pc:	pc:	pc:	pc:	
	prn: 14	prn: 14	prn: 14	prn: 14	prn:	

5	sur: lucy	sur:	sur:	sur:	sur:	sur: square	*absorption*
	noun: (person x)	verb: find	noun: n_1	adj: big	adj: blue	noun: square	*with*
	cat: snp	cat: #n' #a' v	cat: sn' snp	cat: adnv	cat: adnv	cat: sn	*simultaneous*
	sem: nm f	sem: past ind	sem: indef sg	sem: pad	sem: pad	sem: sg	*substitution*
	fnc: find	arg: (p. x) n_1	fnc: find	mdd: n_1	mdd:	fnc:	
	mdr:	mdr:	mdr: big	mdr:	mdr:	mdr:	
	nc:	nc:	nc:	nc: blue	nc:	nc:	
	pc:	pc:	pc:	pc:	pc: big	pc:	
	prn: 14	prn: 14	prn: 14	prn: 14	prn: 14	prn:	

6	sur: lucy	sur:	sur:	sur:	sur:	sur: •	*absorption*
	noun: (person x)	verb: find	noun: square	adj: big	adj: blue	verb: v_1	
	cat: snp	cat: #n' #a' v	cat: sn' snp	cat: adnv	cat: adnv	cat: v' decl	
	sem: nm f	sem: past ind	sem: indef sg	sem: pad	sem: pad	sem:	
	fnc: find	arg: (p. x) square	fnc: find	mdd: square	mdd:	arg:	
	mdr:	mdr:	mdr: big	mdr:	mdr:	mdr:	
	nc:	nc:	nc:	nc: blue	nc:	nc:	
	pc:	pc:	pc:	pc:	pc: big	pc:	
	prn: 14	prn: 14	prn: 14	prn: 14	prn: 14	prn:	

result

sur: lucy	sur:	sur:	sur:	sur:
noun: (person x)	verb: find	noun: square	adj: big	adj: blue
cat: snp	cat: #n' #a' decl	cat: sn' snp	cat: adnv	cat: adnv
sem: nm f	sem: past ind	sem: indef sg	sem: pad	sem: pad
fnc: find	arg: (p. x) square	fnc: find	mdd: square	mdd:
mdr:	mdr:	mdr: big	mdr:	mdr:
nc:	nc:	nc:	nc: blue	nc:
pc:	pc:	pc:	pc:	pc: big
prn: 14	prn: 14	prn: 14	prn: 14	prn: 14

The analysis is (i) *surface compositional* because each lexical item has a concrete sur value and there are no surfaces without a proplet analysis. The derivation order is (ii) *time-linear*, as shown by the stair-like addition of a next word proplet. The activation and application of operations is (iii) *data-driven*.

Each derivation step ‘consumes’ exactly one next word. In each concatenation, the language-dependent sur value provided by lexical lookup is omitted.[1] Lexical lookup and syntactic-semantic concatenation are incrementally intertwined: lookup of a new next word occurs only after the current next word has been processed into the current sentence start.

In a graphical hear mode derivation like 2.1.3, cross-copying between two proplets is indicated by two diagonal arrows and the result is shown in the next line. This includes changes in the cat and the sem slots. For example, the canceling of the n′ (nominative) and a′ (accusative) valency positions in the cat slot of the *find* proplet of lines 2 and 3 is indicated by #-marking.[2]

An absorption is indicated by a single horizontal arrow, indicating the replacement of a substitution variable. With simultaneous substitution the horizontal arrow is joined by diagonal arrows pointing at the variables to be sub-

[1] A partial exception are name proplets, which preserve their sur value in the form of a marker written in lower case default font, e.g. lucy. In the speak mode, the marker is converted back into a regular sur value written in Helvetica, e.g. Lucy.

[2] Canceling by #-marking preserves the canceled value for use in the DBS speak mode. This is in contradistinction to Categorial Grammar (CG), which cancels valency positions by deletion (loss of information). As a sign-based system, CG does not distinguish between the speak and the hear mode.

stituted. For example, the core value of the determiner is the substitution variable n_1. The determiner is modified with the first adnominal *big* (line 3). A second modifier *blue* is conjoined to the first (line 4). Finally, the core value of the noun proplet *square* simultaneously replaces all occurrences of the substitution variable n_1, namely in (i) the determiner, (ii) the first adnominal, and (iii) the verb (line 5). Line 6 shows an absorption without substitution.

2.2 Hear Mode Operations

DBS uses three of kinds of operations: (i) concatenation (2.2.2), (ii) navigation (2.6.3), and (iii) inferencing (7.2.4). They share the following structural properties:

2.2.1 STRUCTURAL PROPERTIES COMMON TO DBS OPERATION KINDS

1. An operation consists of an antecedent, a connective, and a consequent.
2. The antecedent and the consequent each consist of a short list of pattern proplets which are semantically connected by proplet-internal address.
3. Hear- and speak mode operations apply by binding the variables of the antecedent to matching constants of the input, which enables the consequent to derive the output.
4. Inferences apply by matching an input to the antecedent (deductive use, 3.5.1) or the consequent (abductive use, 3.5.2).
5. The codomain of a variable in a pattern proplet may be restricted by an explicit list of possible values (variable restriction, e.g. 5.5.2).

Consider SBJ×PRD concatenating subject and predicate in line 1 of 2.1.2.

2.2.2 CROSS-COPYING *[person x]* AND *find* WITH SBJ×PRD (line 1)

$$\textit{pattern level}\quad \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: NP}\\ \text{fnc:}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{cat: NP}'\text{ X v}\\ \text{arg:}\\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: NP}\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: }\beta\\ \text{cat: \#NP}'\text{ X v}\\ \text{arg: }\alpha\\ \text{prn: K}\end{bmatrix}$$

Variable restriction omitted. See TExer 2.1.2 for ‘Agreement conditions.’

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

$$\textit{content level}\quad \begin{bmatrix}\text{sur: Lucy}\\ \text{noun: }\textbf{[person x]}\\ \text{cat: snp}\\ \text{sem: nm f}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} \begin{bmatrix}\text{sur: found}\\ \text{verb: }\textbf{find}\\ \text{cat: n}'\text{ a}'\text{ v}\\ \text{sem: past ind}\\ \text{arg:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix} \quad \begin{bmatrix}\text{sur: lucy}\\ \text{noun: }\textbf{[person x]}\\ \text{cat: snp}\\ \text{sem: nm f}\\ \text{fnc: }\textbf{find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{verb: }\textbf{find}\\ \text{cat: \#n}'\text{ a}'\text{ v}\\ \text{sem: past ind}\\ \text{arg: }\textbf{[person x]}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}$$

As a hear mode operation, SBJ×PRD is activated if the next word proplet provided by automatic word form recognition matches its second input pattern (data-driven, matching conditions 1.6.1). The activated operation looks at the now front (2.3.2) for a proplet matching its first input pattern and applies.

When a constant matches a corresponding variable, the variable is bound to the constant. For example, the variable in [noun: α] (pattern level) is bound to the 'named referent' value in [noun: [person x]] (content level), and the variable in [verb: β] is bound to the value in [verb: find] ($\Uparrow$). Reusing these variables in the output patterns ($\Rightarrow$) establishes the subject/predicate relation by copying the core value [person x] of the subject proplet into the arg slot of the predicate proplet *find* by replacing the variable β, and the core value of the predicate proplet into the fnc slot of the subject proplet by replacing the variable α ($\Downarrow$).[3]

The hear mode operation to apply next is PRD×OBJ. It is triggered by the proplet of the incoming next word a(n) matching the second input pattern of the operation (data-driven):

2.2.3 Cross-copying *find* and *n_1* with PRD×OBJ (line 2)

PRD×OBJ

pattern level

$$\begin{bmatrix}\text{verb: }\beta\\ \text{cat: }\underline{\#X'}^{4}\ N'\ Y\ \gamma\\ \text{arg: }Z\\ \text{prn: }K\end{bmatrix}\begin{bmatrix}\text{noun: }\alpha\\ \text{cat: }CN'\ N\\ \text{fnc:}\\ \text{prn:}\end{bmatrix}\Rightarrow\begin{bmatrix}\text{verb: }\beta\\ \text{cat: }\underline{\#X'}\ \#N'\ Y\ \gamma\\ \text{arg: }Z\ \alpha\\ \text{prn: }K\end{bmatrix}\begin{bmatrix}\text{noun: }\alpha\\ \text{cat: }CN'\ N\\ \text{fnc: }\beta\\ \text{prn: }K\end{bmatrix}$$

Variable restriction omitted. See TExer 2.3.4 for 'Agreement conditions.'

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

content level

$$\begin{bmatrix}\text{sur:}\\ \text{verb: }\textbf{find}\\ \text{cat: }\#n'\ a'\ v\\ \text{sem: past ind}\\ \text{arg: [person x]}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}\begin{bmatrix}\text{sur: a(n)}\\ \text{noun: }\textbf{n\_1}\\ \text{cat: }sn'\ snp\\ \text{sem: indef sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix}\qquad\begin{bmatrix}\text{sur:}\\ \text{verb: }\textbf{find}\\ \text{cat: }\#n'\ \#a'\ v\\ \text{sem: past ind}\\ \text{arg: [person x] }\textbf{n\_1}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{noun: }\textbf{n\_1}\\ \text{cat: }sn'\ snp\\ \text{sem: indef sg}\\ \text{fnc: }\textbf{find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}$$

The feature [cat: CN′ N] matches a determiner proplet as well as an elementary noun proplet; in the latter case, CN′ would be bound to NIL.

The determiner beginning the phrasal object noun is continued with big:

[3] In DBS, parsing language is for building content from surfaces (hear mode) and deriving surfaces from content (speak mode). Content is built by connecting proplets with the classical semantic relations of structure, coded by address.

This differs from parsing for proving well-formedness, as when using an XML parser on a document marked-up in XML. Early PSG parsers were also limited to deciding the well-formedness of surfaces, but later developments complemented well-formedness judgements with grammatical phrase structure analyses. These, in turn, were semantically interpreted by adding Predicate Calculus.

[4] The underline prevents $\underline{\#X'}$ from being bound to NIL, TExer 1.6.5 (2).

2.2.4 CROSS-COPYING *n_1* AND *big* WITH **DET×ADN** (line 3)

DET×ADN

$$
\begin{array}{lcccccc}
\textit{pattern level} &
\begin{bmatrix}\text{noun: N\_}n\\ \text{cat: CN}'\text{ NP}\\ \text{mdr:}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{adj: }\alpha\\ \text{cat: ADN}\\ \text{mdd:}\\ \text{nc:}\\ \text{prn:}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{noun: N\_}n\\ \text{cat: CN}'\text{ NP}\\ \text{mdr: }\alpha\\ \text{prn:K}\end{bmatrix} &
\begin{bmatrix}\text{adj: }\alpha\\ \text{cat: ADN}\\ \text{mdd: N\_}n\\ \text{nc:}\\ \text{prn: K}\end{bmatrix}
\end{array}
$$

Variable restriction omitted. See TExer 2.2.4.

$$
\begin{array}{lcccccc}
 & & \Uparrow & & & \Downarrow \\
\textit{content level} &
\begin{bmatrix}\text{sur:}\\ \text{noun: }\mathbf{n\_1}\\ \text{cat: sn}'\text{ snp}\\ \text{sem: indef sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} &
\begin{bmatrix}\text{sur: big}\\ \text{adj: }\mathbf{big}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix} & &
\begin{bmatrix}\text{sur:}\\ \text{noun: }\mathbf{n\_1}\\ \text{cat: sn}'\text{ snp}\\ \text{sem: indef sg}\\ \text{fnc:}\\ \text{mdr: }\mathbf{big}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} &
\begin{bmatrix}\text{sur:}\\ \text{adj: }\mathbf{big}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd: }\mathbf{n\_1}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}
\end{array}
$$

The proplet representing a(n) is called *n_1* in the heading of 2.2.4 and 2.2.6 because the name of a proplet (written in *italic* font) equals the core value, here the variable n_1 (and not a language-dependent sur value like a(n)).

The first adnominal modifying the object noun phrase is continued by ADN×ADN, which conjoins the second adnominal *blue* to *big* (15.4.4):

2.2.5 CROSS-COPYING *big* AND *blue* WITH **ADN×ADN** (line 4)

ADN×ADN

$$
\begin{array}{lcccccc}
\textit{pattern level} &
\begin{bmatrix}\text{adj: }\alpha\\ \text{cat: ADN}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{adj: }\beta\\ \text{cat: ADN}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{adj: }\alpha\\ \text{cat: ADN}\\ \text{nc: }\beta\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{adj: }\beta\\ \text{cat: ADN}\\ \text{nc:}\\ \text{pc: }\alpha\\ \text{prn: K}\end{bmatrix}
\end{array}
$$

where ADN ϵ {adn, adnv}

$$
\begin{array}{lcccccc}
 & \Uparrow & & & & \Downarrow \\
\textit{content level} &
\begin{bmatrix}\text{sur:}\\ \text{adj: }\mathbf{big}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd: n\_1}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} &
\begin{bmatrix}\text{sur: blue}\\ \text{adj: }\mathbf{blue}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix} & &
\begin{bmatrix}\text{sur:}\\ \text{adj: }\mathbf{big}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd: n\_1}\\ \text{mdr:}\\ \text{nc: }\mathbf{blue}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} &
\begin{bmatrix}\text{sur:}\\ \text{adj: }\mathbf{blue}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc: }\mathbf{big}\\ \text{prn: 14}\end{bmatrix}
\end{array}
$$

The coordination is intrapropositional (NLC Sect. 8.3) and applies by cross-copying between the nc and the pc slots. As a noninitial conjunct, *blue* has no mdd value, but 'borrows' the modified from the first conjunct *big* when needed (NLC Sect. 8.2).

The phrasal object is completed by adding the common noun square with the absorption operation DET∪CN:

2.2.6 ABSORBING *square* INTO *n_1* WITH DET∪CN (line 5)

DET∪CN

$$\textit{pattern level}\quad \begin{bmatrix}\text{noun: N\_}n\\ \text{cat: X CN}'\text{ NP}\\ \text{sem: Y}\\ \text{prn: K}\end{bmatrix} \quad \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: CN}\\ \text{sem: Z}\\ \text{prn:}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: }\alpha\\ \text{cat: X NP}\\ \text{sem: Y Z}\\ \text{prn: K}\end{bmatrix}$$

Variable restriction omitted. See TExer 2.2.4.

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

$$\textit{content level}\quad \begin{bmatrix}\text{sur:}\\ \text{noun: }\mathbf{n\_1}\\ \text{cat: nn}'\text{ np}\\ \text{sem: def}\\ \text{fnc: find}\\ \text{mdr: big}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{adj: big}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd: }\mathbf{n\_1}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} \begin{bmatrix}\text{sur: square}\\ \text{noun: }\mathbf{square}\\ \text{cat: sn}\\ \text{sem: sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix} \quad \begin{bmatrix}\text{sur:}\\ \text{noun: }\mathbf{square}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: find}\\ \text{mdr: big}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{adj: big}\\ \text{cat: adn}\\ \text{sem: pad}\\ \text{mdd: }\mathbf{square}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}$$

DET∪CN replaces the three occurrences of the variable n_1 (2.1.2, line 5) with the value square by simultaneous substitution.[5]

2.3 A-Memory

A-memory (earlier called word bank) is the main section of the agent's onboard database; it stores content using placeholders for concept values. The auxiliary B-memory provides declarative definitions and procedural operations for placeholders representing complex concepts, e.g. routines. The auxiliary C-memory is the counterpart of B-memory for elementary concepts.

Technically, the A-memory of DBS cognition is realized as a content-addressable database (Sect. 15.6). It stores self-contained content proplets which are concatenated into complex content by proplet-internal address. An address like (square 14) uniquely identifies a proplet by its core and prn value. This suggests a two-dimensional database schema: horizontal for the prn values and vertical for the core values.

2.3.1 TWO-DIMENSIONAL DATABASE SCHEMA OF A-MEMORY

- *horizontal*
  Proplets with the same core value are stored in the same token line in the time-linear order of their arrival.
- *vertical*
  Token lines are in the alphabetical order induced by the letter sequence of their shared core value.

[5] The sentence-final operation applying in line 6, called S∪IP (TExer Sect. 6.3, 18), is omitted.

The time-linear arrival order of the member proplets is reflected by the position in their token line and by their prn value. The (i) member proplets are followed (ii) by a free slot as part of the column called the *now front*, and (iii) the owner.

2.3.2 Schematic example of a token line

$$
\begin{array}{cccc}
\textit{(i) member proplets} & & \textit{(ii) now front} & \textit{(iii) owner} \\
\begin{bmatrix}\text{noun: square}\\ \ldots \\ \ldots \\ \text{prn: 3}\end{bmatrix}
\begin{bmatrix}\text{noun: square}\\ \ldots \\ \ldots \\ \text{prn: 6}\end{bmatrix}
\begin{bmatrix}\text{noun: square}\\ \ldots \\ \ldots \\ \text{prn: 14}\end{bmatrix} & & & \text{square}
\end{array}
$$

The owner equals the core values in the token line and is used for access in storage and retrieval. Proplets provided by current recognition, by A-memory, or by inferencing are stored at the now front in the token line corresponding to their core value. After being assembled into a proposition by matching (data-driven) operations, the now front is cleared by moving it and the owners to the right into fresh memory space (loom-like clearance, Sect. 6.5). This leaves the proplets of the current proposition behind in what is becoming their permanent storage location as member proplets never to be changed, like sediment.

Consider the content assembled in 2.1.2 after now front clearance:[6]

2.3.3 Storing the content derived in 2.1.2 in A-memory

$$
\begin{array}{ccccc}
 & \textit{member proplets} & & \textit{now front} & \textit{owner values} \\
\ldots & \begin{bmatrix}\text{sur:}\\ \text{adj: big}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd: shoe}\\ \text{mdr:}\\ \text{nc: blue}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix} & \begin{bmatrix}\text{sur:}\\ \text{adj: big}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd: square}\\ \text{mdr:}\\ \text{nc: blue}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} & & \text{big} \\
\ldots & \begin{bmatrix}\text{sur:}\\ \text{adj: blue}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd: car}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 5}\end{bmatrix} & \begin{bmatrix}\text{sur:}\\ \text{adj: blue}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc: big}\\ \text{prn: 14}\end{bmatrix} & & \text{blue} \\
\ldots & \begin{bmatrix}\text{sur:}\\ \text{verb: find}\\ \text{cat: \#n}' \text{ \#a}' \text{ decl}\\ \text{sem: pres ind}\\ \text{arg: dog bone}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 8}\end{bmatrix} & \begin{bmatrix}\text{sur:}\\ \text{verb: find}\\ \text{cat: \#n}' \text{ \#a}' \text{ decl}\\ \text{sem: past ind}\\ \text{arg: [person x] square}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} & & \text{find}
\end{array}
$$

. . .	sur: lucy noun: [person z] cat: snp sem: nm f fnc: walk mdr: nc: pc: prn: 11	sur: lucy noun: [person x] cat: snp sem: nm f fnc: find mdr: nc: pc: prn: 14	Lucy
. . .	sur: lucy noun: [person z] cat: snp sem: nm f fnc: walk mdr: nc: pc: prn: 11	sur: lucy noun: [person x] cat: snp sem: nm f fnc: find mdr: nc: pc: prn: 14	[person x]
. . .	sur: noun: square cat: pnp sem: def pl fnc: seek mdr: small nc: pc: prn: 7	sur: noun: square cat: snp sem: indef sg fnc: find mdr: big nc: pc: prn: 14	square

Lucy/[person x] as a name proplet has two proplet instantiations which are stored in different token lines. One uses the surface value, here Lucy, as the owner for access in the hear mode. The other uses the core value, here the 'named referent' [person x], as the owner for access in the speak mode (CTGR). Additional proplets preceding in the token lines are indicated by preceding '. . . '.

The A-memory is content-addressable for two reasons. First, it does not use a separate index (catalog), as in a coordinate-addressable database (Sect. 15.6, CLaTR Sect. 4.1). Second, the 'content' used for a proplet's storage in and retrieval from of the agent's content-addressable memory is the letter sequence of the proplet's core value, which enables string search (Sect. 12.5).

More specifically, for storage in the agent's A-memory, a proplet is written to the now front in the token line of the owner[7] which equals its core value. For declarative retrieval (i.e. not by pointer), the first step goes to the owner corresponding to the sought proplet's core value (vertical) and the second step goes along the token line to the sought proplet's prn value (horizontal).

[6] The B-memory to the right of owners (12.1.1) is omitted.

[7] The terminology of member proplets and owner values is reminiscent of the member and owner records in a classic network database (Elmasri and Navathe (1989[1]–2017[7]), which inspired the database schema of the A-memory in DBS.

2.4 Loomlike Clearance of Now Front

The now front is cleared by moving it and everything to its right (i.e. the owners and B-memory (12.1.1)) into fresh memory territory, leaving its proplets behind in what is becoming their final storage position (Sect. 6.5). This loomlike clearance results in the token lines' time-linear arrival order in A-memory.

Clearance of the current now front is triggered when its proplets have ceased to be candidates for additional concatenations.[8] This is basically the case when an elementary proposition is completed (formally indicated by the automatic incrementation of the prn value for the next proposition (NLC 13.5.1, IP∼START). Exceptions are extrapropositional (i) coordination (NLC Chap. 11) and (ii) functor-argument (NLC Chap. 7; TExer Sects. 2.5, 2.6, 3.3–3.5). In these two cases, the verb of the completed proposition must remain at the now front for cross-copying with the verb of the next proposition until the extrapropositional relation has been established.

Horizontally, the number of proplets in a token line affected by a clearance is either zero or one.[9] Vertically, over the whole column, the number of proplets at the current now front is usually no more than four or five.[10]

The now front made it possible to replace the rule packages of earlier LAG, e.g. NEWCAT, CoL, and TCS, by data-driven application: in the hear mode, a next word proplet (i) is stored in its token line at the now front prior to processing and (ii) activates all operations matching it with their second input pattern; the activated operations (iii) look for a proplet at the now front matching their first input pattern, and (iv) apply if they find one (Sect. 2.2).

2.5 Semantic Relations of Structure

Without nonlanguage cognition there would be no nonlanguage content, and without nonlanguage content there would be no mapping between nonlanguage and language content. The simplest, most straightforward way to enable interaction between language and nonlanguage content is coding them alike by using the same data structure and the same semantic relations of structure.[11] Accordingly, DBS uses (i) the same attributes in the same order for language

[8] A possible neuroscientific parallel to a now front clearance is moving content from short term memory into long term memory (Dubnau et al. 2003).

[9] Except for those rare cases in which several proplets have the same core and prn value, as in Oh Mary, Mary, Mary! or slept and slept and slept. When a now front slot is filled, a new free slot is opened. Thus, the token line of *sleep* would contain three instances of *sleep* proplets with the same prn value at the now front. They will all be left behind during the next loom-like clearance.

[10] For step by step derivations of now front states see CLaTR Sect. 13.3; NLC Sects. 11.2, 11.3.

and nonlanguage proplets, (ii) the same concepts and indexicals as their core and continuation values, and (iii) the same semantic relations of structure for connecting language or nonlanguage proplets into complex content by means of address (Sect. 6.2).

For the graphical characterization of the classical semantic relations of structure, DBS uses / for subject/predicate, \ for object\predicate, | for modifier|modified, and − for conjunct−conjunct. The lines (slashes) are reused in the name of associated operations, whereby the domain and range are characterized by the core attributes noun, verb, and adj, abbreviated as N, V, and A, respectively. Of the 36 potential intrapropositional relations (CLaTR 7.6.1–7.6.3), the following 12 are used in English:

2.5.1 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 to the intrapropositional relations, there are the extrapropositional (clausal) relations (NLC Chaps. 7, 9). Of the 36 potential extrapropositional relations, English uses the following five (CLaTR 7.6.5):

2.5.2 Extrapropositional relations of English

13. V/V subject_clause/matrix_verb
14. V\V object_clause\matrix_verb
15. V|N adnominal_clause|noun (aka relative clause)
16. V|V adverbial_clause|verb
17. V−V extrapropositional verb−verb coordination

[11] The semantic relations of structure are in contradistinction to the semantic relations of the lexicon such as hypernymy-hyponymy, synonymy-antonymy, etc. (Sects. 9.1–9.3).

The clausal argument in an extrapropositional relation is represented by a V in the initial position, i.e. the domain of the relation. The relations 13, 14, 15, and 17 have intrapropositional counterparts, namely 8, 9, 10, and 6, respectively.

Regarding language and nonlanguage cognition, the following differences need to be accommodated. First, the agent's nonlanguage cognition has to deal with a multi-modal environment, while language cognition usually deals with mono-modal surfaces (CLaTR Chap. 8) in the sensory modalities for recognition and action (11.2.1). Second, language interpretation benefits from the fixed time-linear order of the raw data input, while the sequencing in nonlanguage vision, for example, is controlled by moving the agent's viewframe, as when looking around, or when looking at movement within the viewframe, as when watching a moving train. Third, natural language uses function words, morphological variation, and agreement, while nonlanguage recognition presumably deals without these language-dependent differentiations.

At the same time, language cognition borrows the concepts[12] of nonlanguage cognition as core and continuation values, and nonlanguage cognition borrows the semantic relations of structure from language cognition. More specifically, in language cognition, the operations establishing semantic relations between concepts are based (i) on proplets resulting from lexical lookup, (ii) matching these proplets with operation patterns, and (iii) binding variables in proplet patterns to corresponding constants in matching content. In nonlanguage cognition, in contrast, the variables in the operations establishing semantic relations are not bound to concepts, but replaced by them (13.6.1).

2.6 Speak Mode Operations and Laboratory Set-Up

Just as the hear mode is the language part of the agent's recognition, the speak mode is the language part of the agent's action. In the computational reconstruction of the speak mode, three action aspects may be distinguished: the behavior control aspects of (a) *what to say* and (b) *how to say it* (Derr and McKeown 1984; McKeown 1985; Kass and Finin 1988), and (c) the automatic mapping of contents resulting from (a, b) into to language-dependent surfaces.

To temporarily free the linguistic analysis and the software implementation of the speak mode from the behavior control aspects (a) and (b), the output of hear mode interpretation may be used as input to (c) speak mode production.

[12] In analogy to the view in modern physics according to which the meaning of a physical concept is equated with the exact method of measuring it, DBS holds that the content of an elementary concept is determined by the associated recognition and/or action procedure of the agent (grounding).

For example, to produce Lucy found a big blue square in the speak mode, the output of the hear mode derivation 2.1.3 is used as input.

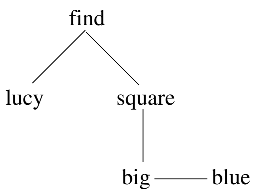
This constellation is called the DBS *laboratory set-up* (TExer Sect. 1.5). It requires the hear mode to automatically derive a content in such a way that its use as input to the speak mode supplies all the grammatical details for reproducing the hearer's input surface as the speaker's output. Reproducing a surface interpreted in the hear mode is an instance of the narrative speak mode.

While the hear mode is driven by a time-linear sequence of incoming external surfaces, the speak mode is driven by the time-linear operations of the think mode, i.e. navigation and inferencing.[13] Thereby an activated current proplet supplies the continuation and prn values (primary key, address) for activating or constructing a successor proplet. This results in an autonomous navigation along existing or newly constructed semantic relations between proplets, which resembles the wandering thoughts or the reasoning of a natural agent.

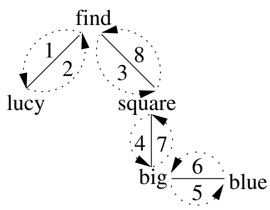
The semantic relations of structure derived in the hear mode interpretation 2.1.3 may be shown equivalently as the following graphs, used as the conceptual basis of the corresponding speak mode production:

2.6.1 Semantic relations underlying speak mode derivation

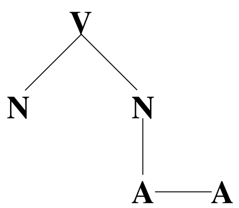
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	3	4	5	6	7	8
Lucy	found	a	big	blue		square	•
V/N	N/V	V\N	N|A	A–A	A–A	A|N	N|V

A kind of line has the same interpretation in all four representations. For example, the / line is the subject/predicate relation[14] no matter whether it is

[13] The fundamental time-linear structure of computational cognition in DBS is in concord with the *Sequential Imperative* proposed by Edmondson (2017) using arguments from philosophy.

[14] The triples in the bottom line of the *(iv) surface realization* characterize semantic relations of structure. For example, N/V is the subject/predicate relation between the Noun Lucy serving as the subject and the Verb find serving as the predicate. In comparison, Greenberg's (1963) typology of SVO,

long, as in the graphs (i), (ii), (iii), or short and of a somewhat different angle (for better formatting in print) in (iv) the surface realization.

The (i) SRG uses the proplets' core *values* as nodes, while the (ii) signature uses the core *attributes*.[15] The (iii) NAG supplies the SRG with numbered arcs which are used by the (iv) surface realization for specifying which surface is realized in the traversal of which arc. A surface is always produced from the goal node of an arc. For example, the surface square is realized by the operation A|N (or rather A↑N[16]) from the goal proplet of arc 7.

The graphs (i) and (ii) provide a static view of the semantic relations of structure. In the corresponding linear notation, the lower node in the graph precedes: subject/predicate in a signature is written as N/V, object\predicate as N\V, adj|noun as A|N, and adj|verb as A|V. In coordination, the linear notation follows the horizontal order in the graph, e.g. A−A.

The graphs (iii) and (iv) show a dynamic view of the semantic relations in that they are used for a navigation from one node to the next. The operation names specify the direction by placing the goal proplet last and adding arrow heads. More specifically, the (i) N/V relation is traversed by the operations V↙N and N↗V, (ii) N\V by V↘N and N↖V, (iii) A|N by N↓A and A↑N, (iv) A|V by V↓A and A↑V, (v) A−A by A→A and A←A, and similarly for (vi) N−N, and intrapropositional (vii) V−V.

The process of automatically navigating along existing semantic relations between proplets is called *selective activation*. The first operation to apply in the surface realization of 2.6.1 is V↙N :

2.6.2 Navigating with V↙N from *find* to *lucy* (arc 1)

$$
\begin{array}{llcl}
 & \textbf{V}\swarrow\textbf{N} & & \\
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} & \begin{bmatrix}\text{verb: } \alpha \\ \text{arg: } \beta\ \text{X} \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{sur: lexnoun}(\hat{\beta})\\ \text{noun: } \beta \\ \text{fnc: } \alpha \\ \text{prn: K}\end{bmatrix} \\
 & \quad\Uparrow & & \quad\Downarrow \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} & \begin{bmatrix}\text{sur:} \\ \text{verb: find} \\ \text{cat: \#n}'\ \text{\#a}'\ \text{decl} \\ \text{sem: past ind} \\ \text{arg: [person x] square} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 14}\end{bmatrix} & & \begin{bmatrix}\text{sur: Lucy} \\ \text{noun: [person x]} \\ \text{cat: snp} \\ \text{sem: nm m} \\ \text{fnc: find} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 14}\end{bmatrix}
\end{array}
$$

SOV, and VSO languages, with S standing for subject and V standing for verb, is a mixing of terms. In DBS, it is avoided by encoding the grammatical role of the N/V notation in the /, and the domain and range of the relation by the core attributes N and V (and accordingly for the other relations). A secondary use of each triple is as the name of the operation which utilizes the relation for navigation.

The surface Lucy is produced by the language-dependent lexicalization rule lexnoun, which sits in the sur slot of the output pattern. It takes the marker lucy in the sur slot of the goal proplet at the content level as input (not shown) and overwrites it with the corresponding sur value Lucy as output (shown).

The think mode operations of selective activation resemble the hear mode operations in that they are defined as patterns, though they are restricted to one input and one output pattern. Inferences, in comparison, are unrestricted with regard to the number of patterns, as shown by 6.6.2 (two patterns), 3.5.1 (four patterns), and 10.4.4 (ten patterns).

The next operation selectively activating the content derived in 2.1.3 is N↗V.

2.6.3 NAVIGATING WITH N↗V FROM *lucy* BACK TO *find* (arc 2)

$$
\begin{array}{llcl}
 & \textbf{N}\nearrow\textbf{V} & & \\
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} & \begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\alpha\\ \text{mdr: Z}\\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{sur: lexverb}(\hat{\alpha})\\ \text{verb: }\alpha\\ \text{arg: \#}\beta\text{ Y}\\ \text{prn: K}\end{bmatrix}\\
 & \multicolumn{3}{l}{\text{Z is NIL, or elementary}^{17}\text{ and \#-marked}}\\
 & \Uparrow & & \Downarrow\\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} & \begin{bmatrix}\text{sur: lucy}\\ \text{noun: [person x]}\\ \text{cat: snp}\\ \text{sem: nm m}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix} & & \begin{bmatrix}\text{sur: found}\\ \text{verb: find}\\ \text{cat: \#n}'\text{ \#a}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: \#[person x] square}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}
\end{array}
$$

Like V↙N in 2.6.2, N↗V navigates along the semantic relation N/V established in the hear mode by SBJ×PRD (2.2.1), but in the opposite (upward) direction. The language-dependent surface is realized in 2.6.3 by lexverb (NLC Sect. 12.4): using the values find and past ind in the goal proplet of 2.6.3, lexverb produces the surface found.[18]

---

[15] The attributes noun, verb, and adj are shown as **N, V**, and **A** to emphasize the equivalence with corresponding operation names, e.g. V↙N or A↑N.

[16] Because the Xfig graphics editor used here does not provide a satisfactory representation of arrows in the linear notation of speak mode operation names, the arrow heads are omitted in the *(iv) surface realization*. Nevertheless, the direction is specified unambiguously by the arc number written directly above in the top line. For more detail see NLC 6.1.4 ff.; CLaTR Chap. 7.

[17] If the adnominal modifier is an elementary coordination, as in big fat blue square, only the first adnominal is written into the mdr slot of the modified noun (NLC 8.3.5). If the modifier is phrasal, as in house on the lake, or clausal, as in dog which barked, it is in postnominal position and a traversal may return to the modified only after completion of the complex modifier(s).

[18] For complete hear and speak mode derivations of 24 linguistically informed examples see TExer Chaps. 2–5. Compared to TExer, the operations 2.6.2 and 2.6.3 are slightly simplified.

In 2.6.1, the arc numbering of the (iii) NAG is called depth first in graph theory (TExer Sect. 1.4). Also, the traversal of the NAG happens to be consecutive – as shown by the arc numbering in the (iv) surface realization. There are, however, grammatical structures such as subject and object gapping for which a breadth first arc numbering is appropriate. And there are constructions for which neither a depth first nor breadth first arc numbering results in a consecutive traversal numbering, as in Perhaps Fido is still sleeping (TExer 3.1.11).

A traversal may be empty in the sense that no surface is produced, e.g. arc 6 in 2.6.1. There may also be multiple realizations, i.e. the production of several word form surfaces in the traversal of a single arc, such as could_be_sleeping (TExer Sect. 2.2). And there may be multiple traversals of a node, as in Whom does John say that Bill believes that Mary loves? (15.1.1, graph on the right; TExer Sect. 5.5).

More than one operation may apply in a derivation step, namely in a suspension compensation (NLC 7.5.1, 11.4.2; TExer 2.5.2, 3.1.2) or a lexical ambiguity. This may result in parallel paths, all but one of which usually ‘die’ after one or two next words (local ambiguity). Otherwise there is a global ambiguity (FoCL Sect. 11.3). This is all complexity-theoretically benign as long as there are no recursive ambiguities, which happily are absent in natural language (FoCL 12.5.7).

3. Content Resonating in Memory

Much of day-to-day behavior is routine and requires no more than activating existing behavior programs. However, when faced with an exceptional situation, the agent's cognition should be able to search memory for previous experiences which happen to be suitable for being adapted to the task at hand.

Sect. 3.1 models reference as a proplet at the now front which activates by address the referred-to proplet in A-memory. This is generalized in Sect. 3.2 from individual referents to resonating contents. Navigating around the neighborhood of a resonating content further extends potentially relevant information (Sect. 3.3). Sect. 3.4 extends adaptive behavior from activating existing content to deriving new content by means of inferencing. Sect. 3.5 shows deductive and abductive use of an inference in reasoning. Sect. 3.6 describes the data-driven triggering of DBS operations.

3.1 Reference as a Purely Cognitive Process

Sign-based analytic philosophy defines reference as a relation between language (referring part) and the world (referred-to part).[1] Agent-based DBS, in contrast, confines reference to nouns (1.5.3, 6.4.7) and distinguishes (1) between referring nouns with and without external surfaces and (2) between referred-to nouns with and without external[2] counterparts. The two distinctions may be characterized by the binary values[3] [±surface] and [±external], whereby [+external] reference is called *immediate*, while [−external] reference is called *mediated* (FoCL 4.3.1).

For example, identifying 'the man with the brown coat' (Quine 1960), with someone seen before, or identifying an unusual building with an earlier language content, e.g. something read in a guide book or heard about, are

[1] Reimer and Michaelson (2014) extend the referring part from language to "representational tokens," which include cave paintings, pantomime, photographs, videos, etc. DBS goes further by generalizing the referring part to content *per se*, i.e. without the need for any cognition-external representation.

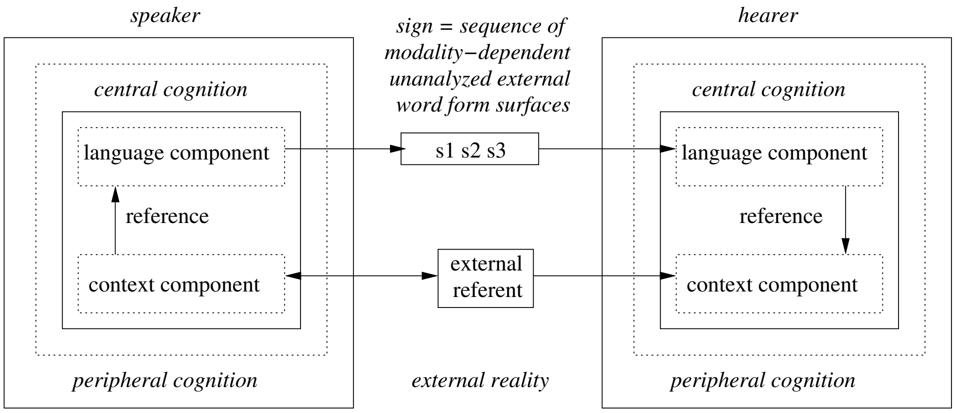
[2] The agent's external surroundings are called the 'task environment' by Newell and Simon (1972).

[3] Binary values are called "feature bundles" by Chomsky and Halle 1968.

[−surface +external]. Talking about Aristotle or J.S. Bach, in contrast, is [+surface −external].

Let us go systematically through the four kinds of generalized DBS reference,[4] beginning with the [+surface +external] constellation between speaker and hearer:

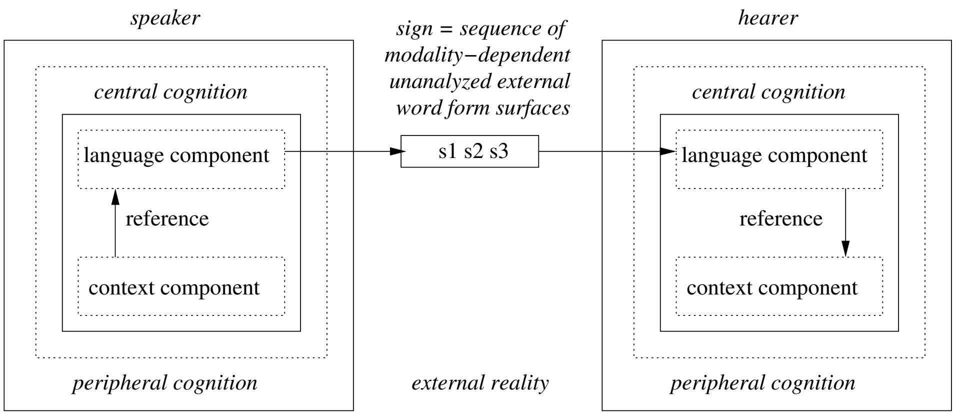
3.1.1 Immediate reference in language communication



Agent-externally, language surfaces (shown here as $\boxed{\text{s1 s2 s3}}$) are modality-specific unanalyzed external signs (raw data) which are passed from the speaker to the hearer and have neither meaning nor any grammatical properties, but may be measured by the natural sciences (Sects. 11.4, 11.5).

The corresponding [+surface −external] constellation between the speaker and the hearer is as follows:

3.1.2 Mediated reference in language communication



[4] The [±surface] and [±external] distinctions are not available in truth-conditional semantics and generative grammar because their sign-based ontology does not provide for different cognitive modes.

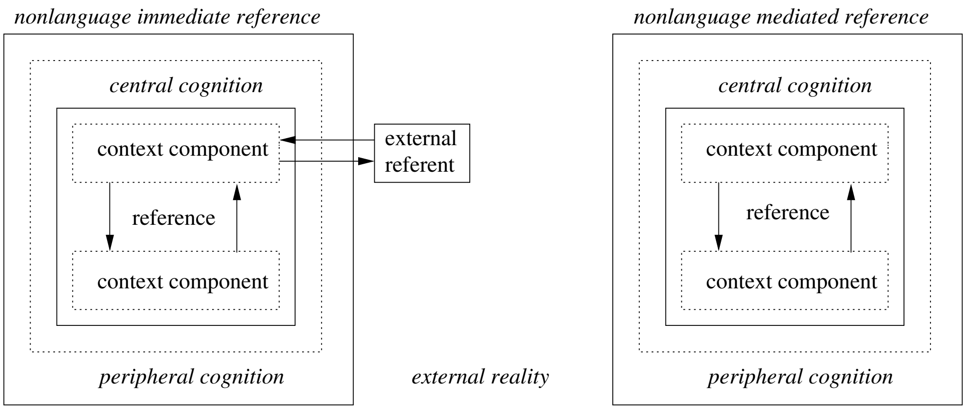
The reference relation begins with content in the memory of the speaker and ends as content in the memory of the hearer. The mechanisms of assigning surfaces to content in the speak mode and content to surfaces in the hear mode are the same in immediate and mediated language reference.[5]

The graphs 3.1.1 and 3.1.2 show the speaker on the left, the sign in English writing order in the middle, and the hearer on the right. This is a possible constellation which is in concord with the naive assumption that time passes with the sun from left to right (→) on the Northern Hemisphere. Yet it appears that the first surface s1 leaves the speaker last and the last surface s3 arrives at the hearer first, which would be functionally incorrect.

It is a pseudo-problem, however, which vanishes if each surface is transmitted individually and placed to the right of its predecessor, i.e. (((s1 s2) s3) s4). This *left-associative*[6] departure and arrival structure allows incremental surface by surface processing, provided the derivation order is based on computing possible continuations, as in Left-Associative Grammar (LAG).

Nonlanguage reference differs from language reference in that it is [−surface]. Thereby nonlanguage immediate reference is [−surface +external] while nonlanguage mediated reference is [−surface −external]:

3.1.3 NONLANGUAGE IMMEDIATE VS. MEDIATED REFERENCE



The referring content in the [−surface +external] constellation is a current nonlanguage recognition. In the [−surface −external] constellation of nonlanguage mediated reference, in contrast, the referring content is activated without an external trigger, for example, by reasoning. In both, the referred-to content is resonating (Sects. 3.2, 3.3) in memory.

[5] On the phone, the speaker may use mediated reference which is immediate for the hearer and vice versa. For example, if the speaker explains to the hearer where to find something in the speaker's apartment, the speaker uses mediated reference and the hearer immediate reference.

[6] Aho and Ullman (1977), p. 47; FoCL 10.1.1.

The schemata 3.1.1–3.1.3 follow the conceptual representation 1.2.2. For the purpose of running software, however, they must be translated into the data structure (1.4.1, 1.4.2) and the database schema (2.3.2, 2.3.3) of DBS.

3.2 Resonating Content: Coactivation by Intersection

For accessing relevant content in memory, the database schema of A-memory supports the software mechanism of coactivation[7] (CLaTR Sect. 5.4). The central notion is the similarity between two contents, one at the current now front, the other resonating in memory (6.5.2–6.5.5). Computationally, contents are similar[8] if they match the same pattern (Sect. 14.1). Degrees of similarity vary with the degree of pattern abstraction as it results from systematically replacing constants with variables (Sect. 14.2).

Coactivation works like a dragnet, pulled by the owners (and corresponding core values) of the proplets currently at the now front. The method is *token line intersection*. The first step is to derive search patterns from concept proplets at the agent's current now front and move them along their respective token lines from right to left (backwards in time).[9]

3.2.1 DERIVING SEARCH PATTERNS FOR A 2ND DEGREE INTERSECTION

$$
\begin{array}{ccc}
\textit{search patterns} & & \textit{proplets at now front} \\
\begin{bmatrix}\text{adj: big}\\ \text{mdd: square}\\ \text{prn: K}\end{bmatrix} & \Leftarrow & \begin{bmatrix}\text{sur:}\\ \text{adj: big}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd: square}\\ \text{mdr:}\\ \text{nc: blue}\\ \text{pc:}\\ \text{prn: 97}\end{bmatrix} \\
 & & \\
\begin{bmatrix}\text{noun: square}\\ \text{mdr: big}\\ \text{prn: K}\end{bmatrix} & \Leftarrow & \begin{bmatrix}\text{sur:}\\ \text{noun: square}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: find}\\ \text{mdr: big}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 97}\end{bmatrix}
\end{array}
$$

7 So far, coactivation as a technical term seems to be confined to muscle action in neurology.

8 Similarity and analogy are being studied in psychology (Gentner and Smith 2012). It appears that such DBS procedures as coactivating resonating content, type/token matching, and binding constants to restricted variables for a computational treatment of similarity and for building content with the semantic relations of structure would be suitable for a computational implementation of these analyses.

In this example, the search patterns express the modifier|modified relation between big and square. As they are moved along their token lines (NLC Sect. 5.1), they retrieve pairs of proplets connected by the same semantic relation and with a shared prn value, as in the following example:[9]

3.2.2 2nd degree intersection coactivating big square

member proplets			*now front*	*owners*
sur: adj: big cat: adn sem: pad mdd: house mdr: nc: red pc: prn: 11	sur: **adj: big** cat: adn sem: pad **mdd: square** mdr: nc: blue pc: **prn: 23**	sur: adj: big cat: adn sem: pad mdd: chair mdr: nc: pc: prn: 32	sur: adj: big cat: adn sem: pad mdd: square mdr: nc: pc: prn: 97	big
sur: **noun: square** cat: snp sem: indef sg fnc: find **mdr: big** nc: pc: **prn: 23**	sur: noun: square cat: snp sem: indef sg fnc: own mdr: green nc: pc: prn: 45	sur: noun: square cat: snp sem: indef sg fnc: buy mdr: red nc: pc: prn: 66	sur: noun: square cat: snp sem: indef sg fnc: find mdr: big nc: pc: prn: 97	square

An intersection of two token lines is of degree 2, of three token lines of degree 3, and so on. Consider extending 3.2.2 into a 3rd degree intersection:

3.2.3 Example of a 3rd degree intersection

member proplets			*now front*	*owners*
sur: adj: big cat: adn sem: pad mdd: house mdr: nc: pc: prn: 11	sur: **adj: big** cat: adn sem: pad **mdd: square** mdr: nc: blue pc: **prn: 23**	sur: adj: big cat: adn sem: pad mdd: chair mdr: nc: pc: prn: 32	sur: adj: big cat: adn sem: pad mdd: square mdr: nc: pc: prn: 97	big
sur: verb: find cat: #n-s3′ #a′ decl sem: past arg: [person y] triangle mdr: nc: pc: prn: 17	sur: **verb: find** cat: #n-s3′ #a′ decl sem: pres arg: [person z] **square** mdr: nc: pc: **prn: 23**		sur: verb: find cat: #n-s3′ #a′ decl sem: past arg: [person x] square mdr: nc: pc: prn: 97	find

9 Moving a search pattern along a token line (NLC 5.1.7) is one of the two standard navigation methods of DBS. The other is moving across token lines by following the semantic relations between proplets.

sur:	sur:	sur:	sur:	
noun: square	noun: square	noun: square	noun: square	square
cat: snp	cat: snp	cat: snp	cat: snp	
sem: indef sg	sem: indef sg	sem: indef sg	sem: indef sg	
fnc: find	fnc: own	fnc: buy	fnc: sell	
mdr: big	mdr: green	mdr: red	mdr: big	
nc:	nc:	nc:	nc:	
pc:	pc:	pc:	pc:	
prn: 23	prn: 45	prn: 66	prn: 97	

Retrieving the 3rd degree intersection from the above A-memory sketch requires the following search patterns:

3.2.4 THREE SEARCH PATTERNS FOR A 3RD DEGREE INTERSECTION

search patterns *proplets at now front*

$$\begin{bmatrix}\text{adj: big}\\ \text{mdd: square}\\ \text{prn: K}\end{bmatrix} \Leftarrow \begin{bmatrix}\text{sur:}\\ \text{adj: big}\\ \text{cat: adnv}\\ \text{sem: pad}\\ \text{mdd: square}\\ \text{mdr:}\\ \text{nc: blue}\\ \text{pc:}\\ \text{prn: 97}\end{bmatrix}$$

$$\begin{bmatrix}\text{verb: find}\\ \text{arg: } \alpha^{10} \text{ square}\\ \text{prn: K}\end{bmatrix} \Leftarrow \begin{bmatrix}\text{sur:}\\ \text{verb: find}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ decl}\\ \text{sem: past}\\ \text{arg: [person x] square}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 97}\end{bmatrix}$$

$$\begin{bmatrix}\text{noun: square}\\ \text{fnc: find}\\ \text{mdr: big}\\ \text{prn: K}\end{bmatrix} \Leftarrow \begin{bmatrix}\text{sur:}\\ \text{noun: square}\\ \text{cat: sn}\\ \text{sem: sg}\\ \text{fnc: find}\\ \text{mdr: big}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 97}\end{bmatrix}$$

Compared to the number of proplets in complete token lines, the number of resonating proplets in an intersection is (i) greatly reduced and (ii) more precisely adapted to the agent's current now front content.[11]

3.3 Resonating Content: Selective Activation

Intersections of higher and higher degree result in a narrowing of the resonating content. This is suitable for finding fewer and fewer but more and more similar precedents.

A different but complementary method is starting a navigation from a low degree intersection by following ‘free’ semantic relations, such as [fnc: find] in the *square* proplet and [nc: blue] in the *big* proplet in 3.2.2. This way of exploring a neighborhood provides the agent with additional contents which exceed the initial intersection, but are likely to be heuristically relevant because they are semantically connected to the initial intersection.

By following free continuation values, the intersection *big square* may be succeeded by a navigation which traverses the complete proposition:

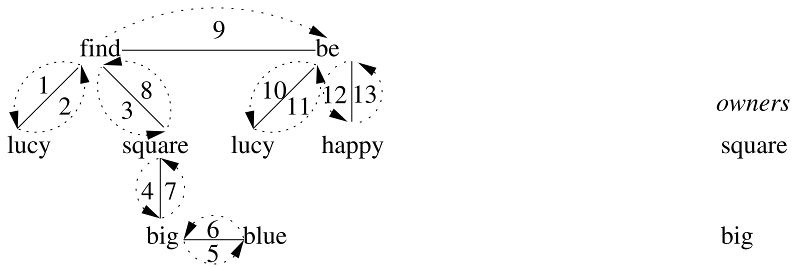
3.3.1 Selective activation following a coactivation

sur: lucy	sur:	sur:	sur:	sur:
noun: **[person x]**	verb: **find**	noun: **square**	adj: **big**	adj: **blue**
cat: snp	cat: #n′ #a′ decl	cat: snp	cat: adnv	cat: adnv
sem: nm f	sem: pres	sem: def sg	sem: pad	sem: pad
fnc: **find**	arg: **[person x] square**	fnc: **find**	mdd: **square**	mdd:
mdr:	mdr:	mdr: **big**	mdr:	mdr:
nc:	nc: (be 24)	nc:	nc: **blue**	nc:
pc:	pc:	pc:	pc:	pc: **big**
prn: 23	prn: 23	prn: 23	prn: 23	prn: 23

The values in bold face are either core or continuation values and specify the semantic relations of structure between the proplets derived in 2.1.3.

Once a coactivation has resulted in the traversal of a complete proposition, it may continue to the next. Consider the semantic relations of the content corresponding to Lucy found a big blue square. She was happy.:

3.3.2 Coactivation of one proposition followed by another



navigation sequence:
big 5 blue 6 big 7 square 8 find 1 lucy 2 find 9 be 10 lucy 11 be 12 happy 13 .

The traversal of the first proposition differs from 2.6.1 in that it begins with a coactivated proplet, i.e. *big*, which is not the top verb. Following (arc 5), the navigation coactivates *blue* and proceeds to *square* (arcs 6 and 7). Using the

[11] For further discussion of autonomous navigation for the coactivation of content see Sect. 13.3.

continuation value find in the fnc slot of *square* as the goal proplet, the navigation continues to the predicate (arc 8). The continuation value [person x] in the arg slot of the predicate, in turn, provides a goal proplet for navigating to the subject (arc 1). The final intrapropositional step is returning to the verb (arc 2), thus completing the traversal of the first proposition.

As indicated by the '−' line in 3.3.2, the extrapropositional relation from the proposition (find 23) to the proposition (be 24) is coordination. The content supporting the extrapropositional coactivation is the following set of proplets:

3.3.3 CONTENT SUPPORTING AN EXTRAPROPOSITIONAL COACTIVATION

sur: lucy	sur:	sur:	sur:	sur:
noun: [person x]	verb: find	noun: square	adj: big	adj: blue
cat: snp	cat: #n′ #a′ decl	cat: snp	cat: adnv	cat: adnv
sem: nm f	sem: pres	sem: def sg	sem: pad	sem: pad
fnc: find	arg: [person x] square	fnc: find	mdd: square	mdd:
mdr:	mdr:	mdr: big	mdr:	mdr:
nc:	nc: **(be 24)**	nc:	nc: blue	nc:
pc:	pc:	pc:	pc:	pc: big
prn: 23	prn: 23	prn: 23	prn: 23	prn: 23

sur: lucy	sur:	sur:	
noun: [person x]	verb: **be**	adj: happy	
cat: snp	cat: #n′ #be′ decl	cat: adn	
sem: nm f	sem: pres	sem: pad	
fnc: be	arg: [person x]	mdd: be	. . .
mdr:	mdr: happy	mdr:	
nc:	nc:	nc:	
pc:	pc:	pc:	
prn: 24	prn: 24	prn: 24	

The extrapropositional coordination is coded by the next conjunct feature [nc: (be 24)] of the predicate *find*.

For a selective activation without language, the navigation sequence in 3.3.2 is possible because the proplets of a content are order-free. For producing a natural language surface, however, a coactivation beginning with *big* and *square* must first go to the predicate *find* in order to traverse the NAG in a way which produces a well-formed surface in the language at hand.

After returning to the top verb, the navigation underlying the speak mode may continue to the next proposition (arc 9) via an extrapropositional coordination (Sect. 13.2), reaching the predicate. The selective activation of DBS is more constrained than Quillian's (1968) *spreading activation* in that it is restricted to a path along existing semantic relations of structure, i.e. functor-argument and coordination, between pairs of content proplets connected by address.

3.4 DBS Inference

A basic distinction in language communication is between the literal meaning_1 of the language expression and the speaker meaning_2 of using the literal meaning_1 in an utterance (PoP-1, 6.1.1). One aspect of an utterance meaning_2 is resonating content in the agent's memory, which was the topic of the preceding Sects. 3.2, 3.3. Another aspect is inferencing, which we turn to now.

Peirce[12] associated the "logical method" of inferencing with three phases of scientific inquiry. *Induction* is the step from a repeated observation of the same correlation, e.g. A & B, to the assumption that if A then B holds in general. However, even if the same correlation has been observed a thousand times, there remains an element of probability: there is no guarantee that the induction if A then B might not fail the next time.

Deduction, in contrast, is based on the form of a correlation. For example,

$$\forall x[f(x) \rightarrow g(x)] \Leftrightarrow \neg\exists x[f(x) \wedge \neg g(x)]$$

is guaranteed to hold if the instantiations of f and g make the premise true, as by setting f = is_rabbit and g = is_furry (but not f = is_turtle and g = is_furry). Set-theoretically, f must be a subset of or equal to g.

Abduction (later also called retroduction by Peirce) is the hypothetical guess at the best explanation. Like induction, it has an element of probability. For example, if Fido seems to always bark when a stranger approaches (3.5.2) and Fido now happens to bark, then it is not unlikely that a stranger is approaching.

In summary, as Peirce (1903) famously put it,

> Deduction proves that something must be; Induction shows that something actually is operative; Abduction merely suggests that something may be.

This characterization is in terms of logical modality, i.e. necessary fact (deduction), contingent fact (induction), and possible fact (abduction).[13]

DBS not only welcomes Peirce's association of inductive, deductive, and abductive reasoning with the three phases of scientific inquiry, but continues with the additional step of tracing the method back to the reasoning of individual cognitive agents (agent-based approach). This is motivated as follows.

[12] As contemporaries with overlapping research areas, especially in their contribution to quantification theory, Peirce (1838–1914) and Frege (1848–1925) raise the question of whether they knew of each other's work; after all, Peirce visited Europe five times between 1870 and 1883 on assignment for the US Geological Service. Hawkins (1993), who investigated the question, did not find a conclusive answer.

[13] The modality of the logical method differs from the sensory (11.2.1) and processing (11.2.2) modalities in that the notion of necessary and possible truth is based on the structure of a logical formula, starting from tautology and contradiction in Propositional Calculus (Sect. 5.3).

A scientific method of inquiry would have little chance of being accepted if it did not concur with the natural reasoning of the scientists and their audience, who are all individual agents. The method must be available to them in principle because otherwise they could not follow, and therefore would not accept, a scientific argument. This is the recognition side of the scientific method in the DBS interpretation.

Conversely, the methods of scientific inquiry are suited best for rational behavior, i.e. behavior designed for maintaining balance and optimizing survival in the agent's ecological niche, whatever it may be. This is the action side of the scientific method in the DBS interpretation.[14]

3.5 Induction, Deduction, Abduction

In the logical method of Peirce, propositions *denote truth values*. In agent-based DBS, propositions *are contents*. DBS codes (i) the determiner quality of the logical quantifiers $\exists$ and $\forall$ as sem and cat values, and (ii) the binding function of the quantifiers by means of prn values (NLC Sect. 6.4).

Peirce uses two different notions of validity: a deduction is valid on the basis of its form, while induction and abduction are stochastic, though with different degrees of probability. DBS, in contrast, views the phases as different aspects of one and the same inference: derived by induction (Sect. 14.3), an inference is used for deduction and abduction. The validity of a DBS deduction (e.g. 3.5.1) is evaluated in terms of the degree in which it supports successful behavior in the agent's ecological niche, i.e. in Darwinian terms, including inferences based on set-theoretic constellations, called S-inferences (Chap. 4).

For example, if the agent observes that Fido seems to always bark when a stranger approaches, this may crystallize into an inference by DBS induction. In deductive use, the input content is matched with the antecedent such that the output content is derived by the consequent:

3.5.1 USING A DBS INFERENCE FOR A DEDUCTION

$$
\begin{array}{llllll}
\textit{pattern level} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{noun: dog}\\ \text{fnc: bark}\\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: dog}\\ \text{prn: K+1}\end{bmatrix}\\
 & \Uparrow & & & & \Downarrow\\
\textit{content level} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: 23}\end{bmatrix} & & \begin{bmatrix}\text{sur: fido}\\ \text{noun: [dog x]}\\ \text{fnc: bark}\\ \text{prn: 24}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: [dog x]}\\ \text{prn: 24}\end{bmatrix}
\end{array}
$$

In abductive use, in contrast, the input content is matched with the consequent such that the output content is derived by the antecedent:

3.5.2 USING THE DBS INFERENCE 3.5.1 FOR AN ABDUCTION

pattern level	noun: stranger fnc: approach prn: K+1	verb: approach arg: stranger prn: K+1		noun: dog fnc: bark prn: K	verb: bark arg: fido prn: K
		⇓			⇑
content level	noun: (stranger 23) fnc: approach prn: 25	verb: approach arg: (stranger 23 prn: 25		sur: fido noun: [dog x] fnc: bark prn: 24	verb: bark arg: [dog x] prn: 24

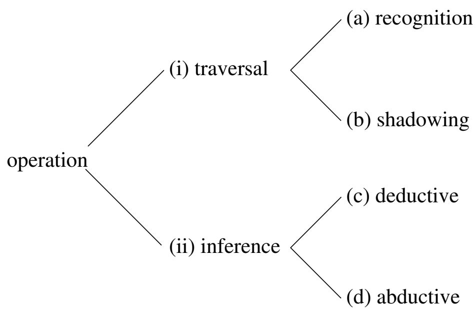
Formally, the abductive use 3.5.2 differs from the deductive use 3.5.1 in the position of ⇑ and ⇓ between the content level and the pattern level. Also, the prn values are adjusted in accordance with the overriding principle that the output of an extrapropositional operation must have a higher prn value than the input (CLaTR 13.5.3).

Going from the consequent to the antecedent by traversing the ⇒ in the opposite direction, upstream so to speak, is indicated by the new connective . It is interpreted as a weakening of the output to the status of a mere hypothesis (in accordance with Peirce). For example, the abductive application in 3.5.2 does not guarantee that a stranger is approaching – Fido may be barking for other reasons (3.6.6).

3.6 Data-Driven Application of Operations

DBS operations are data-driven in that they are activated whenever their respective trigger (Sects. 13.5, 13.6) is supplied with matching input. The four basic operation kinds in a DBS robot may be summarized as follows:

3.6.1 FOUR KINDS OF DATA-DRIVEN OPERATIONS



[14] Providing the autonomous behavior control of a cognitive agent with rational behavior does not preclude the modeling of irrational behavior. Besides defective reasoning, cognition may be led astray by the introduction of irrational beliefs.

The first distinction is between (i) traversal and (ii) inferencing. Traversal operations add one proplet after another into the current sentence start, as in the hear mode, or activate one proplet after another, as in the narrative speak mode. Inferencing takes takes a content as input to a derive new content; it is used for reasoning including the activation or derivation of blueprints for action (CLaTR Chap. 5).

In (i) traversal, there is a secondary distinction between the processing of (a) new proplets provided by recognition, and (b) old proplets provided by shadowing (6.6.2, 6.6.4). In (ii) inferencing, there is a secondary distinction between (c) deductive and (d) abductive use.

The application of data-driven operations is managed by tightly controlling potential input data, i.e. by systematically loading and clearing the now front. As the output of data-driven applications triggers the next operation, there are two kinds of continuation: (1) Sequential Application and (2) Parallel Application. Depending on the available input, the two continuation kinds may be mixed.

3.6.2 Sequential Application Principle (SAP)

> An operation AxB may be followed by ByC or CzB if the output of AxB is matched by the antecedent of ByC or the consequent of CzB, where A, B, C are patterns and x, y, z are connectives.

In a Sequential Application, the output content of operation n provides the input content for operation n+1.[15] As an example consider the following example of drinking from a cup as an action sequence:

3.6.3 Sequential application routine

α pick_up cup	$\Rightarrow$ α raise cup to_mouth
α raise cup to_mouth	$\Rightarrow$ α drink liquid
α drink liquid	$\Rightarrow$ α lower cup
α lower cup	$\Rightarrow$ α set_down cup

The second data-driven continuation kind, i.e. Parallel Application, is an output which triggers several successor operations simultaneously:

[15] For a rough sketch, the data-driven sequential application of operations A⇒B B⇒C C⇒D may be shortened to A⇒B⇒C⇒D by using the consequent of operation n as the antecedent of operation n+1. However, this way of shortening is misleading insofar as the output *pattern* of operation n need not be identical with the input pattern operation n+1; instead it is the output *content* produced by one which must be accepted as the input content of the other (possibility of partially overlapping matching conditions).

3.6.4 Parallel Application Principle (PAP)

> If an output content of operation A matches the input pattern of several operations $B_1...B_n$, they are triggered in parallel.

In data-driven inferencing, parallel application is of two kinds, one triggered by content matching the antecedent, the other by content matching the consequent.

The first kind is illustrated by the following example, which shows six inferences with the same antecedent and different consequents (deduction):

3.6.5 Inferences with the same antecedent, used deductively

antecedent		*consequent*
papa comes home	$\Rightarrow$	papa takes off hat
papa comes home	$\Rightarrow$	papa takes off coat
papa comes home	$\Rightarrow$	papa takes off jacket
papa comes home	$\Rightarrow$	papa takes off tie
papa comes home	$\Rightarrow$	papa takes off shoes
papa comes home	$\Rightarrow$	papa opens a beer

The inferences are maximally concrete (14.2.1, 1) in that the only variables assumed are the prn values. Applied deductively, parallel[16] inferencing results in a conjunction: when papa comes home he takes off his hat and his coat and his jacket, etc., and opens a beer. Because the consequents of 3.6.5 are incompatible, abductive use would result in at most a single application.

Next consider the second kind, namely several inferences with the same consequent and different antecedents. In the following example, six inferences are triggered simultaneously by the content fido barks matching their consequent and deriving different outputs with their different antecedents (abduction):

3.6.6 Inferences with the same consequent, used abductively

antecedent	*consequent*
mama comes home	fido barks
mary comes home	fido barks
suzy comes home	fido barks
peter comes home	fido barks
papa comes home	fido barks
stranger approaches	fido barks

Applied abductively, parallel inferencing results in a disjunction: if fido barks then either mama comes home or mary comes home, etc., or there is a stranger

[16] For practical reasons, simultaneously triggered inferences often apply in sequence.

approaching. Because the antecedents are incompatible, deductive use would result in at most a single application.

Newly derived inferences (Sects. 13.4, 14.5) are integrated automatically into the artificial agent's reasoning. This is because the application of an inference, old or new, is data-driven, i.e. it is triggered by content at the current now front matching its antecedent or its consequent, and may be automatically continued by any inferences matching the current output with their antecedent or their consequent pattern.

Remark on Loss of Information

Viewed in isolation, the transition from an input proplet to the raw data of an output surface may be misconstrued as a deletion of information. In successful communication, however, the meaning$_1$ coded by the speaker into the raw data of a language-dependent surface is recovered by the hearer, based on having learned the language. Thus, no information is lost in this case.

Encoding surfaces into raw data and decoding raw data into surfaces is an essential step in inter-agent language communication. If the raw data are distorted in transit, communication is disturbed or breaks down altogether (Information Theory, Shannon and Weaver 1949). In the construction of a talking robot, the transfer of raw data (3.1.1, 3.1.2) is essential.

4. Logical Reasoning

An early highlight in the Western tradition of logical reasoning are the classical syllogisms of Aristotle (384–322 BC) and their further development by the medieval scholastics.[1] In the modern era, the categorical syllogisms have been based on the intuitions of set theory (Euler, 4.1.3).

The set-theoretic foundation allows the systematic reconstruction of the sign-based classical syllogisms as agent-based DBS inferences. As required for functional equivalence with the natural prototype, the S-inferences (S for set-theory) of logical reasoning and the C-inferences of common sense reasoning are integrated seamlessly (5.2.1, 5.4.1) in the artificial cognition of DBS.

4.1 Overview

A categorical[2] syllogism consists of three parts, called premise 1, premise 2, and the conclusion. This may be shown schematically as follows:

4.1.1 Schematic instantiation of a categorical syllogism

Major premise: all M are P
Minor premise: all S are M
Conclusion: all S are P

M is the middle term, S the subject, and P the predicate. M is shared by the two premises. The respective positions of M are called the alignment.

The three parts of a classical syllogism are restricted to the four categorical judgments, named **A, E, I**, and **O** by the Scholastics:

[1] For a critical review of how the understanding of Aristotle's theory of categorical syllogisms changed over the millenia see Read (2017). For a computational automata and factor analysis see Zhang Yin-sheng and Qiao Xiaodong (2009).

[2] The term *categorical* refers to the strict specification of the Aristotelian syllogisms, especially in their medieval form, such as exactly two premises – one conclusion, middle term not in the conclusion, subject/predicate structure of the three parts, using only the four categorical judgements, etc.

4.1.2 THE FOUR CATEGORICAL JUDGEMENTS

A	universal affirmative	$\forall x\, [\, f(x) \rightarrow g(x)\,]$	all f are g
E	universal negative	$\neg\exists x\, [\, f(x) \wedge g(x)\,]$	no f are g
I	particular affirmative	$\exists x\, [\, f(x) \wedge g(x)\,]$	some f are g
O	particular negative	$\exists x\, [\, f(x) \wedge \neg g(x)\,]$	some f are not g

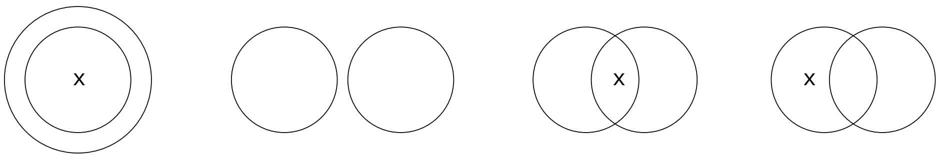
The first-order Predicate Calculus representation in the third column is in a linear notation called *prenex normal form*, which superseded Frege's (1879) graphical format.

The four categorical judgements combine into 256 (2^8) possible syllogism, of which 24 have been found valid. The syllogisms reconstructed in this chapter as DBS inferences are BARBARA,[3] CELARENT, DARII, FERIO, BAROCO, and BOCARDO, plus the modi ponendo ponens and tollendo tollens as special[4] cases.

The set-theoretic constellations underlying the four categorical judgements may be shown as follows:

4.1.3 SET-THEORETIC COUNTERPARTS OF CATEGORIAL JUDGEMENTS

A	E[5]	I	O
$\forall x[f(x) \rightarrow g(x)]$	$\neg\exists x[f(x) \wedge g(x)]$	$\exists x[f(x) \wedge g(x)]$	$\exists x[f(x) \wedge \neg g(x)]$



Known as Euler diagrams[6] (Euler 1761), the set-theoretic constellations are used in DBS to reconstruct the valid syllogisms as data-driven, agent-based, computational S-inferences. As an example, consider the schematic application of modus Barbara in DBS:

[3] The scholastics used the vowels of the categorical judgements in the names of the associated syllogisms as mnemonic support. For example, the three vowels in the name of modus BARBARA indicate that the categorical judgements of the propositions serving as the two premises and the consequent are all of the kind **A**, i.e. universal affirmative (4.1.2).

[4] The modi ponendo ponens (4.2.1) and tollendo tollens (4.3.1) are not categorical syllogism in the narrow sense because their premise 2 and conclusion are not categorical judgements of the kind **A**, **E**, **I**, or **O**. This is reflected by their different naming convention as compared to categorical syllogisms in the narrow sense, for example BARBARA or FERIO.

[5] The categorial judgement E in Predicate Calculus, i.e. (i) $\neg\exists x[f(x) \wedge g(x)\ ]$, is entailed by (ii) $\exists x[f(x) \wedge\neg g(x)\ ]$ and (iii) $\exists x[\neg f(x) \wedge g(x)]$, which have separate set-theoretic counterparts (4.5.8).

4.1.4 Modus Barbara as a DBS inference

$$\begin{array}{lll} \text{inference:} & \alpha \text{ be\_human} \quad \Rightarrow & \alpha \text{ be\_mortal} \\ & \Uparrow & \Downarrow \\ & \text{input: all Greeks be\_human} & \text{output: all Greeks be\_mortal} \end{array}$$

A DBS inference consists of an antecedent pattern, a connective, and a consequent pattern. It takes a content as input and derives a content as output. The purpose of S-inferences is to validly derive new content from given content. The validity follows from set-theoretic intuitions which are the foundation of both the sign-based classical syllogisms and their agent-based counterparts in DBS.

Using modus BARBARA, the transition from a categorial syllogism to a DBS inference may be shown as follows:

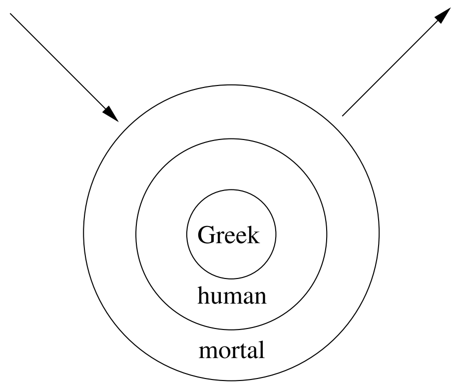
4.1.5 From syllogism to DBS inference

syllogism

premise 1: all Greeks are human
premise 2: all humans are mortal
conclusion: all Greeks are mortal

DBS inference

$$\begin{array}{ll} \alpha \text{ be\_human} \quad \Longrightarrow & \alpha \text{ be\_mortal} \\ \Uparrow & \Downarrow \\ \text{all Greeks be\_human} & \text{all Greeks be\_mortal} \end{array}$$



The DBS reconstruction of a categorical syllogism as an inference has the form α *X implies* α *Y*. The variable α in the antecedent may be matched by and bound to (1) a complete set, e.g. all Greeks (universal, 4.4.4), (2) a subset, e.g. some pets (particular, 4.5.4), or (3) an element, e.g. Socrates (individual, 4.2.3). In the consequent, the input-binding of α derives the output.

[6] Named after Leonhard Euler (1707–1783), the method was known already in the 17th century and has been credited to several candidates.

Venn (1881, p.113) called Euler diagrams “old-fashioned.” Euler diagrams reflect the set-theoretic constellations simple and direct, whereas Venn models the complicated medieval superstructures erected by the scholastics on top of the original syllogisms. Venn diagrams are useful for showing that certain syllogisms, for example, EEE-1 and OOO-1, are not valid.

With the possible presence of negation in the antecedent, the consequent, or both, there result the following four schemata of S-inferences for the categorical syllogisms, each with a universal, a particular, and an individual variant.

The first triple is without negation:

4.1.6 α BE_X IMPLIES α BE_Y

(1) universal: all Greeks be_human implies all Greeks be_mortal.
(2) particular: some pets be_rabbits implies some pets be_furry.
(3) individual: Socrates be_human implies Socrates be_mortal.

The universal version is modeled after modus BARBARA (4.4.1), the particular version after modus DARII (4.5.1), and the individual version after modus ponendo ponens (4.2.1).

The second triple negates the consequent:

4.1.7 α BE_X IMPLIES α NOT BE_Y

(4) universal: all horses be_quadruped implies all horses not be_human.
(5) particular: some pets be_turtles implies some pets not be_furry.
(6) individual: Pegasus be_quadruped implies Pegasus not be_human.

The universal version is modeled after modus CELARENT (4.4.6), the particular version after modus FERIO (4.5.6), and the individual version after modus tollendo tollens (4.3.1).

The third triple negates the antecedent:

4.1.8 α NOT BE_X IMPLIES α BE_Y.

(7) universal: all friars not be_married implies all friars be_single.
(8) particular: some men not be_married implies some men be_single.
(9) individual: Fred not be_married implies Fred be_single.

Set-theoretically, the denotations of not be_married and of be_single are coextensive in all three versions.

The fourth triple negates the antecedent and the consequent. Though EEE syllogisms are not valid for all instantiations, the following instantiations are:

4.1.9 α NOT BE_X IMPLIES α NOT BE_Y

(10) universal: all gods not be_mortal implies all gods not be_human.
(11) particular: some pets not be_furry implies some pets not be_rabbits.
(12) individual: Zeus not be_mortal implies Zeus not be_human.

Set-theoretically, the denotations of not be_X and not be_Y are disjunct in the (10) universal and the (12) individual variant, and in the complement of the pet-rabbit intersection in the (11) particular variant.

4.2 Modus Ponendo Ponens

Modus ponendo[7] ponens serves as the individual version of 4.1.6. The standard representation in Predicate Calculus is as follows:

4.2.1 MODUS PONENDO PONENS IN PREDICATE CALCULUS

$$
\begin{array}{r l}
\text{premise 1:} & \forall x[\mathsf{f}(x) \rightarrow \mathsf{g}(x)] \\
\text{premise 2:} & \exists y[\mathsf{f}(y)] \\
\hline
\text{conclusion:} & \exists z[\mathsf{g}(z)]
\end{array}
$$

Consider the instantiation of f as be_human and g as be_mortal:

4.2.2 INSTANTIATING MODUS PONENDO PONENS

premise 1: For all x, if x is human, then x is mortal.
premise 2: There exists a y, such that y is human.
conclusion: There exists a z, such that z is mortal.

The reconstruction of modus ponendo ponens (NLC Sect. 5.3) in DBS (i) turns premise 1 into the form α is human implies α is mortal, called the inference, (ii) uses premise 2 as the input, and (iii) treats the conclusion as the output:

4.2.3 REPHRASING MODUS PONENDO PONENS IN DBS

inference: α be_human implies α be_ mortal.
input: Socrates be_human.[8]
output: Socrates be_mortal.

Shown here with input for modus ponens (individual), the inference works just as well for particular (4.1.6, 1) and universal (4.1.6, 2) input.

[7] In Propositional Calculus, modus ponendo ponens and modus tollendo ponens differ as follows: modus ponendo ponens has the premises $(A \rightarrow B)$ and A, resulting in the conclusion B; modus tollendo ponens has the premises $(A \vee B)$ and $\neg A$, also resulting in B. The distinction disappears in the functor-argument structure of DBS.

[8] Predicate Calculus treats the copula-adnominal combination is human as the elementary proposition be_human(x) which denotes a truth value. DBS, in contrast, analyzes is human as the modifier|modified (or rather modified|modifier) combination is|human (TExer, Sect. 4.6). For comparison, simplicity, and brevity we compromise here by using the Predicate Calculus notation, e.g. be_human, like intransitive verbs as values in proplets, but without any variable.

Using the DBS data structure, the inference applies as follows to the modus ponendo ponens input of 4.2.3:

4.2.4 Applying modus ponendo ponens as formalized in DBS

$$
\begin{array}{lcccccc}
 & \multicolumn{2}{c}{\textit{antecedent}} & & \multicolumn{2}{c}{\textit{consequent}} \\
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: be\_human} \\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_human} \\ \text{arg: } \alpha \\ \text{prn: K}\end{bmatrix} & \Rightarrow &
\begin{bmatrix}\text{noun: } (\alpha \text{ K}) \\ \text{fnc: be\_mortal} \\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_mortal} \\ \text{arg: } (\alpha \text{ K}) \\ \text{prn: K+1}\end{bmatrix} \\
 & \multicolumn{2}{c}{\Uparrow} & & \multicolumn{2}{c}{\Downarrow} \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{sur: socrates} \\ \text{noun: [person x]} \\ \text{cat: snp} \\ \text{sem: nm m} \\ \text{fnc: be\_human} \\ \text{prn: 23}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_human} \\ \text{cat: \#ns3}' \text{ decl} \\ \text{sem: pres} \\ \text{arg: [person x]} \\ \text{prn: 23}\end{bmatrix} & &
\begin{bmatrix}\text{sur: socrates} \\ \text{noun: [person x]} \\ \text{cat: snp} \\ \text{sem: nm m} \\ \text{fnc: be\_mortal} \\ \text{prn: 23+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_mortal} \\ \text{cat: \#ns3}' \text{ decl} \\ \text{sem: pres} \\ \text{arg: [person x]} \\ \text{prn: 23+1}\end{bmatrix}
\end{array}
$$

The DBS reinterpretation of premise 1 as the inference and premise 2 as the input requires that the input be compatible for matching with the antecedent. This would be prevented, however, if the antecedent specified the noun pattern α as a plural, corresponding to $\forall$x in premise 1 of 4.2.1, and premise 2 as a singular, corresponding to $\exists$y. Therefore, the noun pattern α in 4.2.4 omits the cat and sem features, thus enabling matching (compatibility by omission). By vertically binding the constant socrates of the content level to the variable α in the antecedent of the pattern level, the consequent derives the new content socrates is mortal as output.

4.3 Modus Tollendo Tollens

Modus tollendo[9] tollens serves as the individual version of 4.1.7. A standard representation in Predicate Calculus is as follows:

4.3.1 Modus tollendo tollens in Predicate Calculus

$$
\begin{array}{l}
\text{premise 1: } \forall x[\mathsf{f}(x) \rightarrow \mathsf{g}(x)] \\
\underline{\text{premise 2: } \exists y[\neg \mathsf{g}(y)]} \\
\text{conclusion: } \exists z[\neg \mathsf{f}(z)]
\end{array}
$$

Let us instantiate f as is human and g as is biped:

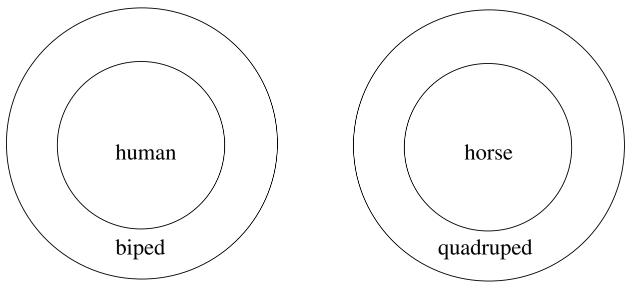
[9] In Propositional Calculus, modus tollendo tollens and modus ponendo tollens differ as follows: modus tollendo tollens has the premises $\neg A \rightarrow \neg B$ and $\neg A$, resulting in the conclusion $\neg B$; modus ponendo tollens has the premises $\neg(A \wedge B)$ and A, also resulting in $\neg B$. The distinction disappears in the functor-argument structure of DBS.

4.3.2 INSTANTIATING MODUS TOLLENDO TOLLENS

premise 1: For all x, if x is human,[10] then x is biped.
premise 2: there exists a y which is not biped.
conclusion: There exists a z which is not human.

If we use quadruped to instantiate not be_biped and horse to instantiate not be_human, the set-theoretic constellation underlying modus tollendo tollens in 4.3.2 may be shown as follows:

4.3.3 SET-THEORETIC VIEW OF MODUS TOLLENDO TOLLENS



Premise 1 corresponds to the set structure on the left, premise 2 to the set structure on the right (with pegasus as the individual instantiation of horse, and the sets horse and human being disjunct).

The DBS reconstruction uses the disjunction of the sets quadruped and human for the inference α be_quadruped implies α not be_human. Pegasus being an element of the set quadruped renders the input Pegasus is quadruped. Pegasus not being an element of the set human renders the output Pegasus not be_human.

4.3.4 REPHRASING MODUS TOLLENDO TOLLENS IN DBS

inference: α be_quadruped implies α not be_human.
input: Pegasus be_quadruped.
output: Pegasus not be_human.

The inference works for the individual, the particular, and the universal variant of 4.1.7; as in 4.3.3, the variants differ solely in their input and output.

Let us conclude with the translation of 4.3.4 into the data structure of DBS:

[10] Predicate Calculus treats the copula-adnominal combination is human as the elementary proposition be_human(x) which denotes a truth value. DBS, in contrast, analyzes is human as the modifier|modified (or rather modified|modifier) combination is|human (TExer Sect. 4.6). For comparison, simplicity, and brevity we compromise here by using the Predicate Calculus notation, e.g. be_human, like intransitive verbs as values in proplets, but without any variable.

4.3.5 APPLYING MODUS TOLLENDO TOLLENS AS FORMALIZED IN DBS

$$
\begin{array}{lcccc}
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} &
\left[\begin{array}{l}\text{noun: }\alpha\\ \text{fnc: be\_quadr.}\\ \text{prn: K}\end{array}\right]
\left[\begin{array}{l}\text{verb: be\_quadr.}\\ \text{arg: }\alpha\\ \text{prn: K}\end{array}\right]
& \Rightarrow &
\left[\begin{array}{l}\text{noun: }(\alpha\text{ K})\\ \text{fnc: be\_human}\\ \text{prn: K+1}\end{array}\right]
\left[\begin{array}{l}\text{verb: be\_human}\\ \text{arg: }(\alpha\text{ K})\\ \text{sem: }\textit{not}\\ \text{prn: K+1}\end{array}\right] \\
& \Uparrow & & \Downarrow \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\left[\begin{array}{l}\text{sur: pegasus}\\ \text{noun: [horse x]}\\ \text{sem: nm m}\\ \text{fnc: be\_quadr.}\\ \text{prn: 23}\end{array}\right]
\left[\begin{array}{l}\text{verb: be\_quadr.}\\ \text{sem: }\\ \text{arg: [horse x]}\\ \text{prn: 23}\end{array}\right]
& &
\left[\begin{array}{l}\text{sur: pegasus}\\ \text{noun: [horse x]}\\ \text{sem: nm m}\\ \text{fnc: be\_human}\\ \text{prn: 23+1}\end{array}\right]
\left[\begin{array}{l}\text{verb: be\_human}\\ \text{sem: }\textit{not}\\ \text{arg: [horse x]}\\ \text{prn: 23+1}\end{array}\right]
\end{array}
$$

The transfer of syllogisms from substitution-driven sign-based symbolic logic to data-driven agent-based DBS relies on DBS inferencing being part of the think mode, which may run detached from the agent's interface component (mediated reference 3.1.3, sequential application 3.6.2).

4.4 Modi BARBARA and CELARENT

The vowels in the name of modus BARBARA indicate the categorical judgements of the propositions serving as the premises and the consequent, which are all of type **A**, i.e. universal affirmative (4.1.2).

4.4.1 MODUS BARBARA IN PREDICATE CALCULUS

$$
\begin{array}{l}
\text{premise 1: } \forall x[\mathsf{f}(x) \rightarrow \mathsf{g}(x)]\\
\underline{\text{premise 2: } \forall y[\mathsf{g}(y) \rightarrow \mathsf{h}(y)]}\\
\text{conclusion: } \forall z[\mathsf{f}(z) \rightarrow \mathsf{h}(z)]
\end{array}
$$

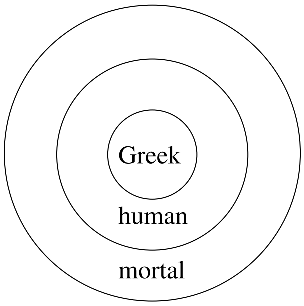
The middle term is g. If f is realized as be_Greek, g as be_human, and h as be_mortal, then the syllogism reads as follows:

4.4.2 INSTANTIATING MODUS BARBARA

premise 1: For all x, if x be_Greek, then x be_human.
premise 2: For all y, if y be_human, then y be_mortal.
conclusion: For all z, if z are Greek, then z be_mortal.

The set-theoretic constellation underlying modus BARBARA in 4.4.2 may be shown as follows:

4.4.3 Set-theoretic view of modus BARBARA



Premise 1 is expressed by the set Greek being a subset of human, and premise 2 by the set Greek being a subset of mortal.

The inference schema of DBS formulates the set-theoretic constellation as follows:

4.4.4 Rephrasing modus BARBARA in DBS

inference:	α be_human implies α be_mortal.
input:	All Greeks be_human.
output:	All Greeks be_mortal.

The validity of the inference follows directly from the subset relations Greek $\subset$ human $\subset$ mortal, which are inherent in the extensions of these concepts.

Consider the translation of 4.4.4 into the data structure of DBS:

4.4.5 Applying modus BARBARA as formalized in DBS

$$
\begin{array}{lllcll}
\begin{array}{l}\textit{pattern}\\\textit{level}\end{array} &
\begin{bmatrix}\text{noun: } \alpha\\ \text{fnc: be\_human}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_human}\\ \text{arg: } \alpha\\ \text{prn: K}\end{bmatrix} & \Rightarrow &
\begin{bmatrix}\text{noun: } (\alpha \text{ K})\\ \text{fnc: be\_mortal}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_mortal}\\ \text{arg: } (\alpha \text{ K})\\ \text{prn: K+1}\end{bmatrix} \\
 & \Uparrow & & & \Downarrow & \\
\begin{array}{l}\textit{content}\\\textit{level}\end{array} &
\begin{bmatrix}\text{noun: greek}\\ \text{cat: pnp}\\ \text{sem: pl exh}\\ \text{fnc: be\_human}\\ \text{prn: 23}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_human}\\ \text{arg: greek}\\ \text{prn: 23}\end{bmatrix} & &
\begin{bmatrix}\text{noun: (greek 23)}\\ \text{cat: pnp}\\ \text{sem: pl exh}\\ \text{fnc: be\_mortal}\\ \text{prn: 23+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_mortal}\\ \text{arg: (greek 23)}\\ \text{prn: 23+1}\end{bmatrix}
\end{array}
$$

By vertically binding greek of the content level to the variable α in the antecedent of the pattern level, the consequent derives the desired new content All Greeks are mortal as output. In the class of syllogisms with unnegated antecedent and unnegated consequent (4.1.6), the reconstruction of BARBARA 4.4.4 constitutes the universal, of DARII 4.5.4 the particular, and of modus ponendo ponens 4.3.4 the individual variant.

Next let us turn to a syllogism with the vowel **E** in its name, where **E** indicates a universal negative (4.1.2). The vowels in the name of modus CELARENT, for example, indicate that premise 1 is of type **E**, premise 2 of type **A**, and the conclusion of type **E**. In Predicate Calculus, CELARENT is represented as follows:

4.4.6 MODUS CELARENT IN PREDICATE CALCULUS

$$\begin{array}{l} \text{premise 1: } \neg\exists x[f(x) \wedge g(x)] \\ \underline{\text{premise 2: } \quad \forall y[h(y) \rightarrow f(y)]} \\ \text{conclusion: } \neg\exists z[h(z) \wedge g(z)] \end{array}$$

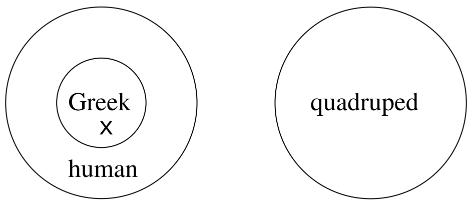
The middle term is f. If f is realized as human, g as quadruped, and h as Greek, then the syllogism reads as follows:

4.4.7 INSTANTIATING CELARENT IN PREDICATE CALCULUS

$$\begin{array}{l} \text{premise 1: } \neg\exists x[\text{human}(x) \wedge \text{quadruped}(x)] \\ \underline{\text{premise 2: } \quad \forall y[\text{Greek}(y) \rightarrow \text{human}(y)]} \\ \text{conclusion: } \neg\exists z[\text{Greek}(z) \wedge \text{quadruped}(z)] \end{array}$$

The set-theoretic constellation underlying modus CELARENT in 4.4.6 may be shown as follows:

4.4.8 SET-THEORETIC VIEW OF MODUS CELARENT



Premise 1 is expressed by the sets human and quadruped being disjunct. Premise 2 is expressed by the set Greek being a subset of human. The conclusion is expressed by the sets Greek and quadruped being disjunct.

The set-theoretic constellation is rephrased as the following inference schema:

4.4.9 REPHRASING MODUS CELARENT IN DBS

inference: α be_human implies α not be_quadruped
input: All Greeks be_human
output: All Greeks not be_quadruped

Consider the translation of 4.4.9 into the data structure of DBS:

4.4.10 APPLYING MODUS CELARENT AS FORMALIZED IN DBS

$$
\begin{array}{lcccccc}
\textit{pattern level} &
\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: be\_human} \\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_human} \\ \text{arg: } \alpha \\ \text{prn: K}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{noun: } (\alpha\ \text{K}) \\ \text{fnc: be\_quadruped} \\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_quadruped} \\ \text{sem: } \textit{not} \\ \text{arg: } (\alpha\ \text{K}) \\ \text{prn: K+1}\end{bmatrix} \\
 & & \Uparrow & & & \Downarrow \\
\textit{content level} &
\begin{bmatrix}\text{noun: greek} \\ \text{cat: pnp} \\ \text{sem: pl exh} \\ \text{fnc: be\_human} \\ \text{prn: 23}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_human} \\ \text{arg: greek} \\ \text{prn: 23}\end{bmatrix} & &
\begin{bmatrix}\text{noun: (greek 23)} \\ \text{cat: pnp} \\ \text{sem: pl exh} \\ \text{fnc: be\_quadruped} \\ \text{prn: 23+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_quadruped} \\ \text{sem: } \textit{not} \\ \text{arg: (greek 23)} \\ \text{prn: 23+1}\end{bmatrix}
\end{array}
$$

By binding greek of the content level to the variable α in the antecedent of the pattern level, the consequent derives the desired new content All Greeks are not quadruped as output. The $\forall$x quantifier of Predicate Calculus is coded in the *greek* proplet by the feature [sem: pl exh] and the negation in the conclusion is coded in the predicate *be_quadruped* by the feature [sem: *not*].

4.5 Modi DARII and FERIO

The DBS variants of modus BARBARA (4.4.4) and modus CELARENT (4.4.9) have shown the treatment of the categorical judgements (4.1.2) **A** (universal affirmative) and **E** (universal negative). To show the treatment of the remaining categorical judgements **I** (particular affirmative) and **O** (particular negative), let us reconstruct the modi DARII and FERIO as DBS inferences.

The vowels in the name DARII indicate the categorical judgment **A** in premise 1, and **I** in premise 2 and the conclusion. The representation in Predicate Calculus is as follows:

4.5.1 MODUS DARII IN PREDICATE CALCULUS

$$
\begin{array}{r}
\text{premise 1: } \forall x[f(x) \rightarrow g(x)] \\
\underline{\text{premise 2: } \exists y[h(y) \wedge f(y)]} \\
\text{conclusion: } \exists z[h(z) \wedge g(z)]
\end{array}
$$

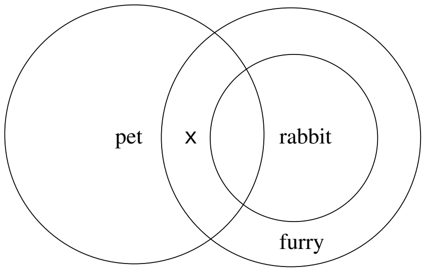
The middle term is f. If f is instantiated as be_rabbit, g as be_furry, and h as be_pet, then the syllogism reads as follows:

4.5.2 INSTANTIATING MODUS DARII

premise 1: For all x, if x is rabbit, then x is furry.
premise 2: For some y, y is pet and y is rabbit.
conclusion: For some z, z is pet and z is furry.

The set-theoretic constellation underlying modus DARII in 4.5.2 may be shown as follows:

4.5.3 SET-THEORETIC VIEW OF MODUS DARII



Premise 1 is expressed by the set rabbit being a subset of furry. Premise 2 is expressed by the set pet overlapping with the set rabbit. The conclusion is expressed by the set pet overlapping with the set furry.

DBS describes the set-theoretic constellation as follows:

4.5.4 REPHRASING MODUS DARII IN DBS

inference: α be_rabbit implies α be_furry.
input: Some pets be_rabbit
output: Some pets be_furry

The inference applies by binding *some pets* in the input to the variable α in the antecedent and using this binding in the consequent to derive the output. The input matches the antecedent and the consequent derives matching output.

Following standard procedure, this is shown in detail by the following translation of 4.5.4 into the data structure of DBS:

4.5.5 APPLYING MODUS DARII AS FORMALIZED IN DBS

$$\begin{array}{lccccc}
\textit{pattern level} & \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: be_rabbit}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: be_rabbit}\\ \text{arg: }\alpha\\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{noun: }(\alpha\text{ K})\\ \text{fnc: be_furry}\\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: be_furry}\\ \text{arg: }(\alpha\text{ K})\\ \text{prn: K+1}\end{bmatrix}\\
 & \Uparrow & & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{noun: pet}\\ \text{cat: pnp}\\ \text{sem: pl sel}\\ \text{fnc: be_rabbit}\\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{verb: be_rabbit}\\ \text{arg: pet}\\ \text{prn: 23}\end{bmatrix} & & \begin{bmatrix}\text{noun: (pet 23)}\\ \text{cat: pnp}\\ \text{sem: pl sel}\\ \text{fnc: be_furry}\\ \text{prn: 23+1}\end{bmatrix} & \begin{bmatrix}\text{verb: be_furry}\\ \text{arg: (pet 23)}\\ \text{prn: 23+1}\end{bmatrix}
\end{array}$$

The particular affirmative quality of the judgement type **I**, i.e. the some, is coded by the features [cat: pnp] and [sem: pl sel] of the *pet* proplets at the content level. Because the grammatical properties of determiners are not reflected at the pattern level (compatibility by omission), modus DARII joins modus BARBARA and modus ponendo ponens as an instance of the DBS inference kind unnegated antecedent and unnegated consequent (4.1.6) in the variant *particular*.

We turn next to modus FERIO. The vowels in the name indicate the categorical judgment **E** (universal negative) in premise 1, **I** (particular affirmative) in premise 2, and **O** (particular negative) in the conclusion. In Predicate Calculus, this is formalized as follows:

4.5.6 MODUS FERIO IN PREDICATE CALCULUS

premise 1: $\neg\exists x[f(x) \wedge g(x)]$
premise 2: $\exists y[h(y) \wedge g(y)]$
conclusion: $\exists z[h(z) \wedge \neg g(z)]$

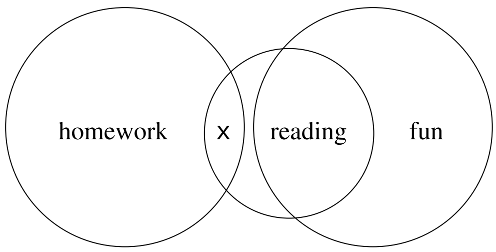
The middle term is g. If f is instantiated as is homework, g as is fun, and h as is reading, then the syllogism reads as follows:

4.5.7 INSTANTIATING MODUS FERIO

premise 1: There exists no x, x is homework and x is fun.
premise 2: For some y, y is reading and y is homework
conclusion: For some z, z is reading and z is no fun

The set-theoretic constellation underlying modus FERIO in 4.5.7[11] may be shown as follows:

4.5.8 SET-THEORETIC VIEW OF MODUS FERIO



Premise 1 is shown by the sets homework and fun being disjunct. Premise 2 is depicted by the sets reading and homework overlapping. The conclusion is shown by the sets reading and homework, and reading and fun overlapping.

[11] Instantiating FERIO with f = dog, g = bird and h = animal also satisfies 4.5.6, but corresponds to a separate set-theoretic constellation (4.1.3).

The inference schema of DBS describes the set-theoretic constellation as follows:

4.5.9 REPHRASING MODUS FERIO IN DBS

inference: α be_homework implies α not be_fun.
input: Some reading be_homework
output: Some reading not be_fun

Consider the translation of 4.5.9 into the data structure of DBS:

4.5.10 APPLYING MODUS FERIO AS FORMALIZED IN DBS

$$
\begin{array}{lccccc}
\textit{pattern level} &
\begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: be\_homew.} \\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_homew.} \\ \text{arg: } \alpha \\ \text{prn: K}\end{bmatrix} &
\Rightarrow
\begin{bmatrix}\text{noun: } (\alpha \text{ K}) \\ \text{fnc: be\_fun} \\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_fun} \\ \text{sem: } \textit{not} \\ \text{arg: } (\alpha \text{ K}) \\ \text{prn: K+1}\end{bmatrix} \\
& & \Uparrow & & \Downarrow \\
\textit{content level} &
\begin{bmatrix}\text{noun: reading} \\ \text{cat: snp} \\ \text{sem: pl sel} \\ \text{fnc: be\_homew.} \\ \text{prn: 23}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_homew.} \\ \text{arg: reading} \\ \text{prn: 23}\end{bmatrix} &
\begin{bmatrix}\text{noun: (reading 23)} \\ \text{cat: snp} \\ \text{sem: pl sel} \\ \text{fnc: be\_fun} \\ \text{prn: 23+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_fun} \\ \text{sem: } \textit{not} \\ \text{arg: (reading 23)} \\ \text{prn: 23+1}\end{bmatrix}
\end{array}
$$

The content noun some reading is characterized by the features [cat: snp] and [sem: pl sel]. The particular negative quality of the judgement type **O** is coded by the feature [sem: *neg*] in the *be_fun* proplets at the pattern as well as the content level. The reconstruction of FERIO in DBS joins the inferences of the kind unnegated antecedent and negated consequent (4.1.7) in the variant *particular*.

4.6 Modi BAROCO and BOCARDO

Like modus FERIO, modus BAROCO has the particular negative **O** in the conclusion. The **A** representing premise 1 indicates the categorical judgment universal affirmative (4.1.2).

4.6.1 MODUS BAROCO IN PREDICATE CALCULUS

premise 1: $\forall x[f(x) \rightarrow g(x)]$
premise 2: $\exists y[h(y) \wedge \neg g(y)]$
conclusion: $\exists z[h(z) \wedge \neg f(z)]$

The middle term is g. If f is instantiated as informative, g as useful, and h as website, then the syllogism reads as follows:

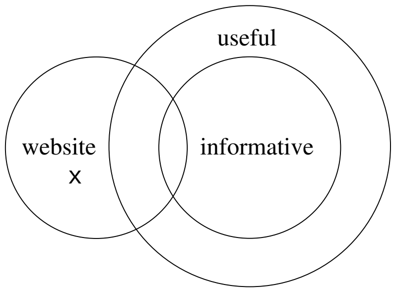
4.6.2 INSTANTIATING MODUS BAROCO

premise 1: All informative things are useful
premise 2: Some website are not informative
conclusion: Some websites are not useful

Among the classical syllogisms, BAROCO is special because the proof of its validity requires a reductio per impossibile.

The set-theoretic constellation underlying modus BAROCO in 4.6.2 may be shown as follows:

4.6.3 SET-THEORETIC VIEW OF MODUS BAROCO



Premise 1 is shown by the set informative being a subset of useful. Premise 2 is depicted by the set website merely overlapping with the set informative. The conclusion is shown by the set website merely overlapping with useful.

The inference schema of DBS describes the set-theoretic constellation as follows:

4.6.4 REPHRASING BAROCO IN DBS

inference: α not be_informative implies α not be_useful
input: Some websites not be_informative
output: Some websites not be_useful

Consider the translation of 4.6.4 into the data structure of DBS:

4.6.5 BAROCO IN DBS

$$
\begin{array}{lcccccc}
\textit{pattern level} &
\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: be\_informative}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_informative}\\ \text{sem: }\textit{not}\\ \text{arg: }\alpha\\ \text{prn: K}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{noun: }(\alpha\text{ K})\\ \text{fnc: be\_useful}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_useful}\\ \text{sem: }\textit{not}\\ \text{arg: }(\alpha\text{ K})\\ \text{prn: K+1}\end{bmatrix} \\
 & & \Uparrow & & & \Downarrow \\
\textit{content level} &
\begin{bmatrix}\text{noun: website}\\ \text{cat: pnp}\\ \text{sem: pl sel}\\ \text{fnc: be\_informative}\\ \text{prn: 25}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_informative}\\ \text{sem: }\textit{not}\\ \text{arg: website}\\ \text{prn: 25}\end{bmatrix} &
 &
\begin{bmatrix}\text{noun: website}\\ \text{cat: pnp}\\ \text{sem: pl sel}\\ \text{fnc: be\_useful}\\ \text{prn: 24}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_useful}\\ \text{sem: }\textit{not}\\ \text{arg: website}\\ \text{prn: 24}\end{bmatrix}
\end{array}
$$

The reconstruction of BAROCO in DBS joins inferences of the kind negated antecedent and negated consequent (4.1.9) in the variant *particular*. It differs from DARII (4.5.4) in that the input and output of BAROCO are negated, while those of DARII are not.

Like modus BAROCO, modus BOCARDO has the particular negative **O** in the conclusion. They differ in that the letters **A** and **O** in premises 1 and 2 are interchanged.

4.6.6 Modus BOCARDO in Predicate Calculus

$$\begin{array}{l} \text{premise 1: } \exists x[f(x) \wedge \neg g(x)] \\ \underline{\text{premise 2: } \forall y[f(y) \rightarrow h(y)]} \\ \text{conclusion: } \exists z[h(z) \wedge \neg g(z)] \end{array}$$

The middle term is f. If f is instantiated as be_cat, g as has_tail, and h as be_mammal, then the syllogism reads as follows:

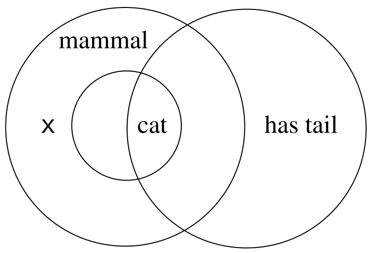
4.6.7 Instantiating modus BOCARDO

premise 1: some cats have no tail
premise 2: all cats are mammals
conclusion: some mammals have no tail

The reductio per impossibile, which helps to prove the validity of BAROCO, is complemented in BOCARDO by ekthesis (Aristotle, An. Pr. I.6, 28b20–21).[12]

The set-theoretic constellation underlying modus BOCARDO in 4.6.7 may be shown as follows:

4.6.8 Set-theoretic view of modus BOCARDO



Premise 1 is the cat complement of the cat and has_tail intersection. Premise 2 is shown by cat being a subset of mammal. The conclusion is shown as the mammal complement of the mammal and has_tail intersection.

[12] In the middle ages, several jails in England, one specifically in Oxford, were called Bocardo because it was so hard for students to learn how to verify this syllogism.

4.6.9 Rephrasing BOCARDO in DBS

inference:	α be_cat implies some α not have_tail.
input:	some mammals are cats
output:	some mammals have no tail

Consider the translation of 4.6.8 into the data structure of DBS:

4.6.10 Applying BOCARDO as a DBS inference

$$
\begin{array}{lccccc}
\textit{pattern level} &
\begin{bmatrix}\text{noun: } \alpha\\ \text{fnc: be\_cat}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_cat}\\ \text{arg: } \alpha\\ \text{prn: K}\end{bmatrix} & \Rightarrow &
\begin{bmatrix}\text{noun: } (\alpha\ \text{K})\\ \text{sem: pl sel}\\ \text{fnc: has\_tail}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: has\_tail}\\ \text{sem: } \textit{not}\\ \text{arg: } (\alpha\ \text{K})\\ \text{prn: K+1}\end{bmatrix} \\
 & & \Uparrow & & & \Downarrow \\
\textit{content level} &
\begin{bmatrix}\text{noun: mammal}\\ \text{cat: pnp}\\ \text{sem: pl sel}\\ \text{fnc: be\_cat}\\ \text{prn: 24}\end{bmatrix} &
\begin{bmatrix}\text{verb: be\_cat}\\ \text{arg: mammal}\\ \text{prn: 24}\end{bmatrix} & &
\begin{bmatrix}\text{noun: mammal}\\ \text{cat: pnp}\\ \text{sem: pl sel}\\ \text{fnc: has\_tail}\\ \text{prn: 25}\end{bmatrix} &
\begin{bmatrix}\text{verb: has\_tail}\\ \text{sem: } \textit{not}\\ \text{arg: cat}\\ \text{prn: 25}\end{bmatrix}
\end{array}
$$

The subject in the consequent pattern of the BOCARDO inference is some α, a restriction which is coded by the feature [sem: pl sel], in contradistinction to the subject of the consequent pattern of the FERIO inference (4.5.10), which is unrestricted and thus compatible with universal, particular, and individual input (compatibility by omission)

Summary

The valid syllogisms and the corresponding DBS inferences are alike in that they are founded on set-theoretic relations between concepts. They differ in that the concepts of the syllogisms rely on the semantic intuitions of the native speakers alone, whereas the corresponding DBS concepts complement these intuitions with procedures which map between declarative definitions provided by the agent's memory and raw data provided by sensors and activators of the agent's interface component.

In this way, artifical cognition is 'grounded,' which supports not only objective testing by running software, but provides a talking robot with real recognition and action. The procedures complement the set-theoric relations with a multitude of additional semantic properties, such as wave-length and frequency in the definition of color concepts (1.3.5, 1.3.6), or length and angle in the definition of two-dimensional geometric concepts (1.3.2, 1.3.3). The sign-based approaches, in contrast, are limited to set-theory as the only (metalanguage independent) semantic structure available.

5. Common Sense Reasoning

Given that logical and common sense reasoning are not separate in the natural prototype, there have been attempts to combine their artificial counterparts into one system. Several past attempts tried to assimilate common sense reasoning to substitution-driven sign-based logical reasoning, for example by using Predicate Calculus and its truth conditions as a convenient semantic notation (Fikes et al. 1971, Nilsson 1988, Hobbs et al. 1993).[1]

The reconstruction of the classical categorical syllogisms as set-theory-based S-inferences in the preceding chapter has shown, however, that logical reasoning is susceptible to a data-driven agent-based approach. The present chapter shows that common sense reasoning (C-inferences) is susceptible to the agent-based approach as well, thus allowing the seamless integration of S- and C- inferences into one coherent system of computational cognition in DBS.

5.1 Logical vs. Common Sense Reasoning

The set-theoretic S-inferences and the common sense C-inferences of DBS have in common that they are derived automatically from content, apply data-driven, have similar antecedent-consequent patterns, and use the same data structure. They differ in that the standard of success for an S-inference is being valid, while the standard of success for a C-inference is maintaining a state of balance. Compare the following two examples:

5.1.1 Example of an S-inference (FERIO, 4.5.10)

S-inference:	α is homework	$\Rightarrow$	α is no fun
	$\Uparrow$		$\Downarrow$
input:	some reading is homework	output:	some reading is no fun

[1] Alternative approaches are Sowa et al. (2003), who take a conceptual graph approach and treat logical reasoning as a highly stylized form of analogical reasoning, and Pei Wang (2013), who presents a non-axiomatic logic founded in part on inheritance.

If the prejudice against homework[2] expressed by the instantiation of the underlying syllogism (4.5.7) is accepted, the output is valid (4.5.9).

The source of a common sense reasoning, in contrast, may be the repeated observation of an imbalance, as in the following example:

5.1.2 Example of a C-inference (CAUSE_AND_EFFECT)

$$\begin{array}{rlcrl}
\text{C-inference:} & \alpha \text{ is hungry} & \Rightarrow & & \alpha \text{ is cranky (5.1.4)} \\
 & \Uparrow & & & \Downarrow \\
\text{input:} & \text{Laura is hungry} & & \text{output:} & \text{Laura is cranky}
\end{array}$$

The formal similarity between S- and C-inferences is illustrated by the following translation of the C-inference into the data structure of DBS:

5.1.3 Applying the C-inference 5.1.2

$$\begin{array}{lcccccc}
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{noun: } \alpha\\ \text{fnc: be_hungry}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: be_hungry}\\ \text{arg: } \alpha\\ \text{prn: K}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{noun: } (\alpha \text{ K})\\ \text{fnc: be_cranky}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: be_cranky}\\ \text{arg: } (\alpha \text{ K})\\ \text{prn: K+1}\end{bmatrix} \\
 & \Uparrow & & & \Downarrow & \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{sur: laura}\\ \text{noun: [girl x]}\\ \text{cat: snp}\\ \text{sem: nm f}\\ \text{fnc: be_hungry}\\ \text{prn: 24}\end{bmatrix} &
\begin{bmatrix}\text{verb: be_hungry}\\ \text{arg: [girl x]}\\ \text{prn: 24}\end{bmatrix} &
 &
\begin{bmatrix}\text{sur: laura}\\ \text{noun: [girl x]}\\ \text{cat: snp}\\ \text{sem: nm f}\\ \text{fnc: be_cranky}\\ \text{prn: 25}\end{bmatrix} &
\begin{bmatrix}\text{verb: be_cranky}\\ \text{arg: [girl x]}\\ \text{prn: 25}\end{bmatrix}
\end{array}$$

The S-inference 4.5.10 and the C-inference 5.1.3 differ in the source of their reasoning. In the S-inference, the source is the disjunction between the concepts homework and be_fun and the intersection between reading and homework, which are assumed to be generally accepted. In the C-inference, in contrast, the source is a personally observed behavior.[3]

Another difference between S- and C-inferences is the possibility of a countermeasure in a C-inference, for which the set-theoretical foundation of an S-inference provides no room. Consider the following example:

5.1.4 Example of a C-inference (COUNTERMEASURE)

$$\begin{array}{rlcrl}
\text{C-inference:} & \alpha \text{ is cranky} & \Rightarrow & & \alpha \text{ needs food} \\
 & \Uparrow & & & \Downarrow \\
\text{input:} & \text{Laura is cranky} & & \text{output:} & \text{Laura needs food}
\end{array}$$

The translation of 5.1.4 into the data structure of DBS is shown as follows:

---

[2] Many children at the elementary level do their homework with pride.

[3] The resulting set-theoretic relation between being cranky and being hungry, i.e. intersection, is merely a consequence of the reasoning, and not the source.

5.1.5 APPLYING THE C-INFERENCE 5.1.4

$$
\begin{array}{lllllll}
\textit{pattern level} & \begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: be\_cranky} \\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: be\_cranky} \\ \text{arg: } \alpha \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{noun: } (\alpha \text{ K}) \\ \text{fnc: need} \\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: need} \\ \text{arg: } (\alpha \text{ K}) \text{ food} \\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{noun: food} \\ \text{fnc: need} \\ \text{prn: K+1}\end{bmatrix} \\
 & & \Uparrow & & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{sur: laura} \\ \text{noun: [girl x]} \\ \text{cat: snp} \\ \text{sem: nm f} \\ \text{fnc: be\_cranky} \\ \text{prn: 24}\end{bmatrix} & \begin{bmatrix}\text{verb: be\_cranky} \\ \text{arg: [girl x]} \\ \text{prn: 24}\end{bmatrix} & & \begin{bmatrix}\text{sur: laura} \\ \text{noun: [girl x]} \\ \text{cat: snp} \\ \text{sem: nm f} \\ \text{fnc: need\_food} \\ \text{prn: 25}\end{bmatrix} & \begin{bmatrix}\text{verb: need} \\ \text{arg: [girl x] food} \\ \text{prn: 25}\end{bmatrix} & \begin{bmatrix}\text{noun: food} \\ \text{cat: snp} \\ \text{sem: nm f} \\ \text{fnc: need} \\ \text{prn: 25}\end{bmatrix}
\end{array}
$$

Finding and applying a countermeasure is a form of problem solving in common sense reasoning (CLaTR Sect. 5.5).

5.2 Combining S- and C-Inferencing

Functional equivalence (Sects. 1.1, 15.2) at a certain level of abstraction between the human prototype and the artificial agent requires computational cognition to apply S- and C-inferencing in one and the same train of thought. Consider the following derivation of a data-driven countermeasure, which begins with the C-inferences 5.1.2 and 5.1.4, continues with a lexical S-inference coding a hypernymy (5.2.2), and concludes with another C-inference:

5.2.1 MIXING S- AND C-INFERENCE IN A TRAIN OF THOUGHT

$$
\begin{array}{lllll}
1 & \text{C-inference:} & \alpha \text{ is hungry} & \Rightarrow & \alpha \text{ is cranky (5.1.2)} \\
 & & \Uparrow & & \Downarrow \\
 & \text{input:} & \text{Laura is hungry} & \text{output:} & \text{Laura is cranky} \\
2 & \text{C-inference:} & \alpha \text{ is cranky} & \Rightarrow & \alpha \text{ needs food (5.1.4)} \\
 & & \Uparrow & & \Downarrow \\
 & \text{input:} & \text{Laura is cranky} & \text{output:} & \text{Laura needs food} \\
3 & \text{S-inference:} & \text{Laura eats } \beta & \Rightarrow & \beta\ \varepsilon\ \{\text{apple, banana, cookie,} \ldots, \text{strawberry}\} \\
 & & \Uparrow & & \Downarrow \\
 & \text{input:} & \text{Laura eats food} & \text{output:} & \text{Laura eats apple or cookie} \\
 & & & & \text{or banana, ..., or strawberry ...} \\
4 & \text{C-inference:} & \alpha \text{ eats cookie} & \Rightarrow & \alpha \text{ is agreeable again} \\
 & & \Uparrow & & \Downarrow \\
 & \text{input:} & \text{Laura eats cookie} & \text{output:} & \text{Laura is agreeable again}
\end{array}
$$

The S-inference 3 illustrates a lexical alternative to the syllogisms analyzed in Chap. 4, namely a hypernymy, which is defined as follows.

5.2.2 Lexical S-inference implementing hypernymy

[noun: α] $\Rightarrow$ [noun: β]
If α is animal, then β ε {ape, bear, cat, dog, ...}
If α is food, then β ε {apple, banana, cookie, ..., strawberry}
If α is fuel, then β ε {diesel, gasoline, electricity, hydrogen, ...}
If α is plant, then β ε {bush, cactus, flower, grass, tree, vine, ...}
...

The set-theoretic structure of a hypernymy[4] is the relation between a superordinate term and its extension. Accordingly, food is the hypernym of apple, banana, cookie, ..., and strawberry. Set-theoretically, the denotation of food equals the codomain of α. The restrictions on variables are species-, culture-, and even agent-dependent. and may be approximated empirically by means of DBS corpus analysis (RMD[5] corpus).

As another example of a lexical S-inference consider the synonymy[6] between abstract and summary:[7]

5.2.3 Lexical S-inference for abstract-summary synonymy

$$\begin{bmatrix}\text{verb: }\alpha\\ \text{arg: X abstract}\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{noun: abstract}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}\ \textbf{syn}\ \begin{bmatrix}\text{verb: }\alpha\\ \text{arg: (X K) summary}\\ \text{prn: K+1}\end{bmatrix}\begin{bmatrix}\text{noun: summary}\\ \text{fnc: (}\alpha\text{ K)}\\ \text{prn: K+1}\end{bmatrix}$$

where α ε {write, read, discuss, ...}

The restriction on the variable α specifies likely verbs, which may be obtained from a corpus database (CLaTR Sect. 15.5).

As the opposite of synonymy consider the following example of an antonymy:

5.2.4 Lexical S-inference implementing an antonymy

$$\begin{bmatrix}\text{noun: }\alpha\\ \text{mdr: good}\\ \text{prn: K}\end{bmatrix}\ \begin{bmatrix}\text{adj: good}\\ \text{mdd: }\alpha\\ \text{prn: K}\end{bmatrix}\ \textbf{ant}\ \begin{bmatrix}\text{noun: (}\alpha\text{ K)}\\ \text{mdr: bad}\\ \text{prn: K+1}\end{bmatrix}\ \begin{bmatrix}\text{adj: bad}\\ \text{mdd: }\alpha\\ \text{sem: }\textit{not}\\ \text{prn: K+1}\end{bmatrix}$$

---

[4] For the corresponding hyponymy see 9.1.1.

[5] Reference-Monitor corpus structured into Domains (CLaTR Sect. 15.3).

[6] Whether or not two concepts are synonymous is notoriously difficult to prove. In DBS, two concepts are regarded as synonymous if they behave the same pragmatically (i.e. pick out the same referents, trigger the same inferences) relative to the full range of interpretation contexts provided to the agent by its ecological niche.

[7] The pairs abstract/summary (5.2.3) and handsome/good-looking (5.2.5) are listed as synonyms at `http://www.englisch-hilfen.de/en/words/synonyms.htm`. Intended to help in writing, the website uses a weak notion of synonymy, namely as possible word alternatives in certain contexts, and not in the strict sense of identical meaning in all contexts.

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.[8] Set-theoretically, the denotations of good and of bad are disjunct.

The following example of a lexical S-inference establishes the synonymy relation between handsome and good-looking:

5.2.5 LEXICAL S-INFERENCE FOR handsome-good_looking SYNONYMY

$$\begin{bmatrix}\text{verb: be}\\ \text{mdr: handsome}\\ \text{prn: K}\end{bmatrix} \ \textbf{syn} \ \begin{bmatrix}\text{verb: be}\\ \text{mdr: good-looking}\\ \text{prn: K+1}\end{bmatrix}$$

Set-theoretically, good-looking and handsome are coextensive.

5.3 Validity and Balance

There is a general understanding that the term *valid* applies to an inference if the consequent follows correctly from the antecedent. The term *truth*, in contrast, applies to individual propositions. Thereby arises the choice between treating truth (i) as an elementary, atomic value, as used in Propositional Calculus[9] for unanalyzed, atomic propositions, or (ii) as a derived notion for analyzed, contingent proposition tokens (6.2.2, 6.2.4), which indicates whether the agent's cognition worked correctly or not.

The unanalyzed atomic propositions of Propositional Calculus are represented by a letter like A. Their unanalyzed, atomic truth values are based on the intuition that $(A \vee \neg A)$ is always true (tautology), while $(A \wedge \neg A)$ is always false (contradiction). In a two-valued system, this logical aspect of truth is represented by the atomic values true (also written as $\top$, 1, T) and false (also written as $\perp$, 0, F).

For a substitution-driven sign-based approach like Predicate Calculus, there is no agent and therefore no option of defining contingent truth in terms of cognition working correctly or not. Instead, Predicate Calculus tries to analyze the inside of propositions by reusing the atomic truth values and the operators of Propositional Calculus. For example, the contingent proposition Lucy found a big blue square, derived in 2.1.3 in the DBS hear mode as a set of order-free proplets connected by address, has the following analysis in Predicate Calculus:

[8] The equivalence between a concept and its negated antonym is usually questionable. For example, when Bruckner specifies the tempo of the scherzo in his Sixth Symphony as nicht schnell (not fast), he did not mean langsam (slow).

[9] For a reconstruction of Propositional Calculus in DBS with special attention to Boolean satisfiability (SAT problem) see Hausser 2003.

5.3.1 Lucy found a big blue square IN PREDICATE CALCULUS

$$\exists x[\text{find}'(\text{Lucy}', x) \wedge [\text{big}'(x) \wedge \text{blue}'(x) \wedge (\text{square}'(x))]]$$

The transitive verb find, the adns big and blue, and the noun square are represented uniformly as the mini-propositions find′(Lucy′, x), big′(x), blue′(x), and square′(x), which denote truth values relative to an arbitrary model and are connected by the $\wedge$ operator of Propositional Calculus. The variable x is bound by the quantifier $\exists$(x).

For demonstrating the interpretation explicitly, a model or model structure (1.2.3-1.2.5) may be defined for each constant, e.g. F(blue′) = $\{a_0, a_1, a_2, a_3, a_4, a_5\}$ and F(square′) = $\{a_4, a_5, a_6, a_7, a_8\}$. The subformula [blue′(x) $\wedge$ (square′(x)] is true relative to this model because the intersection of the two denotation sets is defined to be non-empty, and would otherwise be false.

Instead of reusing the atomic truth values $\top$ and $\perp$ of Propositional Calculus for truth relative to a model, as in sign-based Predicate Calculus, agent-based DBS uses a contingent notion of truth. For example, if the agent observes that all children in a given context of interpretation are asleep and communicates this fact by saying all children are sleeping, it is speaking truly. Semantically, the sentence asserts that there is a set of more than one child (pl) and all elements of the set (exh) participate in whatever is asserted by the verb (NLC 6.4.7).

DBS uses the propositional truth values $\top$ and $\perp$ for defining the truth conditions of the propositional operators $\vee, \wedge, \rightarrow$, and $\neg$, and combines them with the notion of *balance* (NLC Sect. 3.5; CLaTR Sect. 5.1, 5.2). While the propositional truth values are static in that $\neg$A (A is false) is the opposite of A (A is true) without any system-internal possibility of turning $\neg$A into A, the balance principle (CLaTR 5.1.2, MoC-4) is dynamic in that $\approx$A (A is out of balance) may[10] provide an impulse to get back to $\div$A (A is balanced).

The DBS combination of the truth operators $\top$ and $\perp$ of Propositional Calculus and the balance operators $\approx$A and $\div$A is based on the following formulas, which adapt the laws of De Morgan:

5.3.2 EQUIVALENCES BASED ON TRUTH AND BALANCE OPERATORS

$$\approx A \Leftrightarrow \neg(\div A)$$
$$\div A \Leftrightarrow \neg(\approx A)$$
$$\neg(\approx(A \wedge B)) \Leftrightarrow (\neg(\approx A) \vee \neg(\approx B)) \Leftrightarrow (\div A \vee \div B)$$
$$\neg(\approx(A \vee B)) \Leftrightarrow (\neg(\approx A) \wedge \neg(\approx B)) \Leftrightarrow (\div A \wedge \div B)$$

[10] If an opponent O is out of balance, as in wrestling, however, agent A may want to increase the imbalance.

The method for turning a current state of imbalance into a state of balance[11] is data-driven DBS inferencing (CLaTR Sect. 5.5): in common sense reasoning, C-inferences take precedents as input and derive blueprints for action as output. Their purpose is to return the agent into state of balance.

5.4 Analogy

Common sense reasoning is based on relations provided by repeated observation and contingent knowledge. For example, there is nothing law-like or set-theoretic in Laura being cranky when hungry. There is another dimension, however, namely analogy: a truck not starting caused by a lack of fuel may be seen as analogous to being cranky caused by a lack of food. The following train of thought is modeled after 5.2.1:

5.4.1 COMMON SENSE REASONING BASED ON ANALOGY

1	C-inference:	α has no fuel	$\Rightarrow$	α does not start (5.4.2)
		$\Uparrow$		$\Downarrow$
	input:	truck has no fuel	output:	truck does not start
2	C-inference:	α does not start	$\Rightarrow$	α needs fuel
		$\Uparrow$		$\Downarrow$
	input:	truck does not start	output:	truck needs fuel
3	S-inference:	truck gets β	$\Rightarrow$	β ϵ {diesel, gasoline, electricity,...}
		$\Uparrow$		$\Downarrow$
	input:	truck gets fuel	output:	truck gets diesel or gasoline or electricity or hydrogen...
4	C-inference:	α gets fuel	$\Rightarrow$	α starts
		$\Uparrow$		$\Downarrow$
	input:	truck gets fuel	output:	truck starts

As in 5.2.1, the C-inference 1 is a general common sense observation, while the C-inference 2 applies to a particular instance. The lexical S-inference 3 is an instance of the hypernymy 5.2.2, while the C-inference 4 derives the desired result.

Using the data structure of DBS, the data-driven application of the second inference in 5.4.1 may be shown as follows:

[11] The drive to maintain a state of balance may be viewed as the DBS counterpart to the notion of intention as used by Grice (1975). For discussion see CLaTR Sect. 5.1.

5.4.2 APPLYING THE C-INFERENCE 2 OF 5.4.1

$$
\begin{array}{lcccccc}
\textit{pattern level} &
\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: start}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: start}\\ \text{sem: }\textit{not}\\ \text{arg: }\alpha\\ \text{prn: K}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{noun: }(\alpha\text{ K})\\ \text{fnc: need}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: need}\\ \text{arg: }(\alpha\text{ K})\text{ fuel}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{noun: fuel}\\ \text{cat: snp}\\ \text{fnc: need}\\ \text{prn: K}\end{bmatrix}
\end{array}
$$

where α ε {motor cycle, car, truck, tank,...}

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

$$
\begin{array}{lcccccc}
\textit{content level} &
\begin{bmatrix}\text{noun: truck}\\ \text{cat: snp}\\ \text{fnc: start}\\ \text{prn: 24}\end{bmatrix} &
\begin{bmatrix}\text{verb:start}\\ \text{sem: }\textit{not}\\ \text{arg: truck}\\ \text{prn: 24}\end{bmatrix} &
&
\begin{bmatrix}\text{noun: truck}\\ \text{cat: snp}\\ \text{fnc: need}\\ \text{prn: 25}\end{bmatrix} &
\begin{bmatrix}\text{verb: need}\\ \text{arg: (truck 24) fuel}\\ \text{prn: 25}\end{bmatrix} &
\begin{bmatrix}\text{noun: fuel}\\ \text{cat: snp}\\ \text{fnc: need}\\ \text{prn: 25}\end{bmatrix}
\end{array}
$$

Finding and applying analogical countermeasures may be based in part on a systematic development of semantic fields (11.3.3) across domains.

5.5 Hierarchy Formation

The descriptive power of the DBS C-inferencing method may be shown further by condensing a complex content into a meaningful summary. The short text used in the following example is explicitly 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.

A reasonable summary of the content would be *car accident*. This summary may be represented as the content in the following sketch of an A-memory:

5.5.1 TEXT SUMMARIZED AS car accident AND STORED IN A-MEMORY

	member proplets		*now front*	*owner values*
...			[noun: accident mdr: (car 1) prn: 67]	accident
...	[noun: car fnc: hit prn: 1]	[noun: (car 1) fnc: speed prn: 2]	[noun: (car 1) mdd: accident prn: 67]	car
...	[verb: hit arg: car tree prn: 1] ...			hit
...		[verb: speed arg: (car 1) prn: 2] ...		speed
...	[noun: tree fnc: hit prn: 1] ...			tree

The connection between propositions 1 and 2 and the summary 67 is coreference by address (Sect. 6.5): the address (car 1) is coded proplet-internally in proposition 2 and in the summary 67, and refers to the initial *car* proplet in proposition 1. The summary consists of the proplets *car* and *accident*, which are connected by the modifier|modified relation and the shared prn value 67.

The summary-creating operation deriving the new content may be defined formally as the following C-inference:

5.5.2 SUMMARY-CREATING C-INFERENCE

$$
\begin{array}{llcl}
 & \textit{antecedent} & & \textit{consequent} \\
\textit{pattern level:} &
\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: hit}\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{verb: hit}\\ \text{arg: }\alpha\ \beta\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: }\beta\\ \text{fnc: hit}\\ \text{prn: K}\end{bmatrix}
& \textbf{sum} &
\begin{bmatrix}\text{noun: }(\alpha\text{ K})\\ \text{mdd: accident}\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{noun: accident}\\ \text{mdr: }(\alpha\text{ K})\\ \text{prn: K+1}\end{bmatrix} \\
 & \multicolumn{3}{l}{\text{where } \alpha\ \varepsilon\ \{\text{car, truck, boat, ship, plane, ...}\} \text{ and } \beta\ \varepsilon\ \{\text{tree, rock, wall, mountain, ...}\} \cup \alpha} \\
\textit{matching and binding} & \Uparrow & & \Downarrow \\
\textit{content level:} &
\begin{bmatrix}\text{noun: car}\\ \text{fnc: hit}\\ \text{prn: 1}\end{bmatrix}
\begin{bmatrix}\text{verb: hit}\\ \text{arg: car tree}\\ \text{prn: 1}\end{bmatrix}
\begin{bmatrix}\text{noun: tree}\\ \text{fnc: hit}\\ \text{prn: 1}\end{bmatrix}
& &
\begin{bmatrix}\text{noun: (car 1)}\\ \text{mdd: accident}\\ \text{prn: 67}\end{bmatrix}
\begin{bmatrix}\text{noun: accident}\\ \text{mdr: (car 1)}\\ \text{prn: 67}\end{bmatrix}
\end{array}
$$

The variable restriction specifies α as a vehicle like a car or a truck and β as an obstacle like a tree, a rock, or another car. There is no set-theoretic relation between the input pattern and the output pattern. The connective is **sum**(marize).

The possible values which α and β may be bound to during matching are restricted by the codomains of these variables: the variable α generalizes the summary-creating inference to different kinds of accidents, i.e. *car accident, truck accident,* etc., while the 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 argument 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 data stored in A-memory may never be modified.

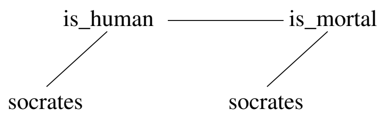
Summarizing content into shorter and shorter versions may be seen in the context of semantic networks in general and semantic hierarchies in particular, which are intensely researched in today's science. The emerging hierarchies provide retrieval relations for upward or downward traversal. Upward traversal supplies the agent with more general notions at a higher level of abstraction, while downward traversal supplies more concrete instantiations at a lower level. Either kind may be used for (a) accessing and applying inferences defined at another level of abstraction, and for (b) subsequently returning to the original level (CLaTR Sect. 6.5).

5.6 Navigating vs. Inferencing

In the DBS think mode, the semantic relations in a content may be used (i) for navigating and (ii) for inferencing. In navigating, existing relations are traversed by selective activation (Sect. 3.3). In inferencing, the antecedent (deduction) or the consequent (abduction) is activated by matching content and used to derive new relations as output to the now front (CLaTR Sect. 13.5). Furthermore, navigating and inferencing alike may be either without or with an accompanying realization in a natural language of choice.

As an example, let us compare the use of modus ponendo ponens (Sect. 4.2) (i) for a narrative speak mode realization using existing content, and (ii) for reasoning, i.e. for deriving new content. Both use the same static semantic relations of structure for traversal:

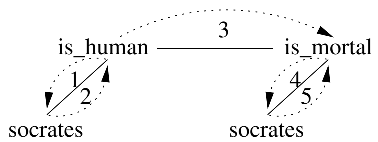
5.6.1 SEMANTIC STRUCTURE UNDERLYING MODUS PONENDO PONENS



The subject/predicate relations between (i) socrates and be_human and (ii) socrates and be_mortal correspond to the signature[12] N/V. The coordination connecting the two propositions represented by be_human and be_mortal corresponds to the signature V−V.

The use of the static semantic structure dynamically for (i) navigating and (ii) inferencing may be shown as the following NAG,[13] which extends 5.6.1 by adding numbered arcs:

5.6.2 NAVIGATIONAL ASPECT OF MODUS PONENDO PONENS



The navigation steps are be_human↙socrates (arc 1), socrates↗be_human (arc 2), be_human→be_mortal (arc 3), be_mortal↙socrates (arc 4), and socrates↗be_mortal (arc 5).

The difference between (i) navigation and (ii) inference is in the interpretation of arc 3. In navigation, arc 3 is interpreted as licensing the traversal of an

[12] Cf. (ii) in 2.6.1.
[13] Cf. (iii) in 2.6.1.

existing relation between two existing contents (recognition). In inferencing, in contrast, arc 3 is interpreted as an instruction to build new content from an existing input as an output (action). In this respect, S- and C-inferences are alike.

The difference between the think mode (a) with and (b) without accompanying language surfaces depends on whether the agent 'has language,' and if so, decides to turn on the lexicalization rules (2.6.2 2.6.3) in the sur slot of the proplets which constitute the activated or derived content. A precondition for successful language communication is the presence of a speaker and a hearer speaking the same language and connected by an open channel of communication.

Part II

Pragmatics

6. STAR-0 Content: Origin

Modern philosophy of language analyzes meaning mostly as either (i) the truth-conditions of symbolic logic (e.g. Davidson 1967, Montague 1973, Kamp 1980) or as (ii) the use-conditions of ordinary language philosophy (e.g. Austin [1955]1962, Grice 1957, Searle 1969). Though fundamentally different, they are alike insofar as they are not agent-based. For example, even though Grice's (1957) definiendum U meant something by uttering x mentions the utterer U, it takes the perspective of an outside observer (CLaTR 5.1.1).

DBS, in contrast, takes the perspective of an agent looking, listening, and sniffing into the world. For the agent, the world is given in the form of raw data. What is not recognized in the raw data does not externally exist for the agent. What is not realized by the agent as raw data does not exist for others.

6.1 Context of Interpretation

In natural language communication, two kinds of meaning must be distinguished: (a) the literal meaning_1 of a language sign and (b) the speaker meaning_2 of an utterance in which the sign's literal meaning_1 is being used (Hausser 1980 et seq.):

6.1.1 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 notions of meaning_1 and meaning_2 belong to language cognition. They are generalized in DBS to language and nonlanguage cognition alike by subsuming meaning_1 under content type and meaning_2 under content token:

6.1.2 PoP-1^+ as a generalization of PoP-1

> The agent's use of a content type relative to an agent-internal context of interpretation is a content token.

For building a talking robot, PoP-1 and PoP-1^+ raise the following question:

6.1.3 Fundamental question of using a content type

What exactly is the computational nature of the context of interpretation?

In the narrow sense, the context of interpretation is the current state of the agent's on-board orientation system, represented as a list of parameter features called the STAR. In a wider sense, it consists of those contents stored in the agent's entire A-memory which happen to automatically resonate with the current now front content (Sect. 3.2). In addition there is exploring the neighborhood of a resonating content by autonomous navigation (Sect. 3.3).

A STAR is a special kind of proplet which has the following attributes:

6.1.4 STAR attributes S, T, A, R, 3rd, and prn

- S specifies the agent's current location in Space.
- T specifies the agent's current moment in Time.
- A specifies the Agent her-, him-, or itself (pro1).
- R specifies the intended Recipient (partner of discourse, pro2).
- 3rd specifies items which are neither first nor second person (pro3).
- prn takes a number as value (6.2.2, 6.2.4)

The letters of the last two attributes are omitted in the acronym for the sake of a simple, memorable abbreviation.

The S value is provided by landmarks like a big rock or a street sign, but may also be GPS data radio-transmitted directly into computational cognition, bypassing the agent's interface component. The T value is provided by changes between day and night or a clock (CLaTR Sect. 14.2), but may likewise be radio-transmitted directly into the robot's cognition. The A value is the agent's name; in any given agent, it serves as the anchor of perspective and does not change (11.6.1).[1] The R and 3rd values result from the interaction with other agents or items. The prn value, finally, is assigned jointly to the STAR and the associated content by the agent's on-board orientation system.

As an alert agent monitors the current situation, there results a sequence of contents (3.3.3). In language use (pragmatics[2]) each content is connected to the current STAR by means of a shared prn value. Depending on the agent's perspective, there are altogether three kinds, called STAR-0, STAR-1, and STAR-2. They consist of the same list of attributes, but differ in some of the values they take.

[1] Such self-reference is essential for recognizing oneself in a mirror (mirror test, Gallup 1970).

[2] For a more traditional, truth-conditional approach to pragmatics see Groenendijk and Stokhof (1978).

STARs are attached to nonlanguage and language content alike.[3] In other words, the think mode processes content regardless of whether the proplets happen to have sur(face) values or not. The sole purpose of language-dependent, unanalyzed, external surfaces and the associated word form production and recognition is transfer of content from the speaker to the hearer (3.1.1, 3.1.2).

6.2 Four Kinds of Content

The meaning$_1$ of a language sign is a content type, while the utterance meaning$_2$ is a content token. Language types are defined as contents without a STAR, while language tokens are defined as contents connected to a STAR.

Because the type-token distinction applies also to nonlanguage content there are four kinds: nonlanguage content (i) type and (ii) token, and language content (iii) type and (iv) token. The following examples illustrate the four kinds with the same content, i.e. The dog found a bone.

6.2.1 Nonlanguage content type

sur:	sur:	sur:
noun: dog	verb: find	noun: bone
cat: snp	cat: #n′ #a′ decl	cat: snp
sem: def sg	sem: past ind	sem: indef sg
fnc: find	arg: dog bone	fnc: find
mdr:	mdr:	mdr:
nc:	nc:	nc:
pc:	pc:	pc:
prn: K	prn: K	prn: K

This is a content type because there is no STAR and the prn value is a variable, here K, and it is a nonlanguage content because the sur slots are empty.

Next consider the corresponding nonlanguage token:

6.2.2 Nonlanguage content token

sur:	sur:	sur:	S: yard
noun: dog	verb: find	noun: bone	T: Friday
cat: snp	cat: #n′ #a′ decl	cat: snp	A: sylvester
sem: def sg	sem: past ind	sem: indef sg	R:
fnc: find	arg: dog bone	fnc: find	3rd:
mdr:	mdr:	mdr:	prn: 12
nc:	nc:	nc:	
pc:	pc:	pc:	
prn: 12	prn: 12	prn: 12	

The three content proplets and the STAR proplet are connected by a common

[3] Treating language and nonlanguage cognition alike is in concord with empirical results in behavioral psychology (Friedrich et al. 2015; Lech et al. 2016; Mayring 2000, 2007; Stangl 2007) and ethology (Zentall et al. 2008), according to which category formation does not require the language ability.

prn value, here the constant 12. According to the STAR, the token resulted as an observation by the agent Sylvester on Friday in the yard.

The following language content type corresponds to 6.2.1. The example illustrates the independence of language-dependent sur values, here German, from the relatively language-independent placeholders, which use English base forms for convenience:

6.2.3 LANGUAGE CONTENT TYPE (MEANING$_1$)

sur: der_Hund	sur: fand	sur: einen_Knochen
noun: dog	verb: find	noun: bone
cat: snp	cat: #n′ #a′ decl	cat: snp
sem: def sg	sem: past ind	sem: indef sg
fnc: find	arg: dog bone	fnc: find
mdr:	mdr:	mdr:
nc:	nc:	nc:
pc:	pc:	pc:
prn: K	prn: K	prn: K

A language content type is an abstraction because concrete time-linear surface-compositional DBS hear mode derivations, e.g. 2.1.3, result in content tokens. A content type is obtained from a content token by removing the STAR and replacing the prn constants with suitable variables.

Finally consider a language content token which uses the type 6.2.3. It is an utterance meaning$_2$ produced by sylvester in German towards the intended hearer speedy and corresponds to the nonlanguage content token 6.2.2:

6.2.4 LANGUAGE CONTENT TOKEN (MEANING$_2$)

sur: der_Hund	sur: fand	sur: einen_Knochen	S: yard
noun: dog	verb: find	noun: bone	T: Friday
cat: snp	cat: #n′ #a′ decl	cat: snp	A: sylvester
sem: def sg	sem: past ind[4]	sem: indef sg	R: speedy
fnc: find	arg: dog bone	fnc: find	3rd:
mdr:	mdr:	mdr:	prn: 12
nc:	nc:	nc:	
pc:	pc:	pc:	
prn: 12	prn: 12	prn: 12	

The content types 6.2.1 and 6.2.3 match not only the tokens 6.2.2 and 6.2.4, but an open-ended number of similar tokens.

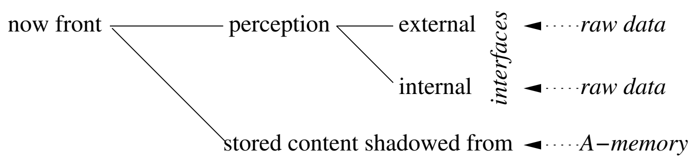
6.3 On-Board Orientation System: STAR-0

The on-board orientation system of a DBS robot is part of the interface component (11.1.1). It assigns the same prn value[5] to (i) the current content and (ii)

[4] The value ind stands for indicative, in contradistinction to progressive and subjunctive.

its STAR of origin, called STAR-0. The recognition side of STAR-0 monitoring has three sources: (i) external (red), (ii) internal (thirsty), and (iii) stored content in and shadowed from A-memory to the now front:

6.3.1 Structure of DBS recognition



Internal perception, for example, takes pro1 as subject, the verb concept feel in the present indicative, and one of the adverbial modifiers in the variable restriction set {angry, bold, bored, cold, dumb, excited, furious, greedy, happy, hot, hungry, lucky, ..., peaceful, stupid, thirsty, tired, ...}:[6]

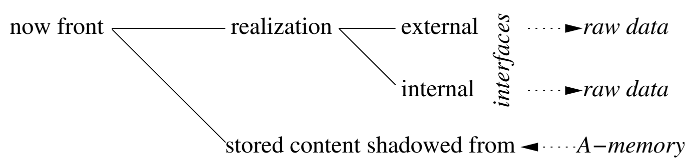
6.3.2 Internal recognition of I feel hungry. bound to STAR-0

noun: pro1	verb: feel	adj: hungry	S: kitchen
fnc: feel	sem: pres ind	mdd: feel	T: 20080915T155300
prn: 22	arg: pro1	prn: 22	A: sylvester
	mdr: hungry		R:
	prn: 22		3rd:
			prn: 22

The content is anchored to the current STAR-0 by a shared prn value, here 22. The S value specifies kitchen as the agent's present location, the T value 20080915T155300 as the current time, and the A value sylvester as the permanent fix point of the agent's on-board orientation system.

The action side of STAR-0 monitoring has three sources as well: (i) external realization (walking), (ii) internal realization (holding one's breath), and (iii) stored action content shadowed from A-memory:

6.3.3 Structure of DBS action



[5] In systems which are run without an on-board orientation system, such as an experimental DBS hear grammar, e.g. TExer Sect. 6.3, the proplets of a content receive their prn value from the concatenation operations, intra- (e.g. 2.2.1) and extrapropositionally (e.g. NLC 12.3.4).

[6] For grammatical constructions in which the verb takes a modifier as the second argument, as in I feel hungry, see TExer Sect. 4.6.

The STAR-0 proplets assigned to recognition and action contents all have the same A-value (11.6.1), namely that of the agent (except perhaps for agents with a multiple personality disorder). The STAR-0s are stored in a single token line in the order of their arrival, whereby the A value serves (i) as the unique owner and (ii) as the core value.[7]

6.3.4 STAR-0 PROPLETS IN A TOKEN LINE

				member proplets		*now front*	*owner value*
	S: kitchen		S: yard	S: living room	S: garden		
	T: 20080915T155300		T: $t_{2+1}-t_4$	T: $t_{4+1}-t_6$	T: $t_{5+1}-t_7$		
…	A: sylvester	…	A: sylvester	A: sylvester	A: sylvester		sylvester
	R:		R: speedy	R: tweety	R: hector		
	3rd:		3rd:	3rd: Speedy	3rd:		
	prn: 22		prn: 71–78	prn: 79–82	prn: 83–87		

In addition to recording the agent's on-board orientation, the STAR-0 proplets of origin serve the interpretation of indexicals contained in the associated content. The spatial indexical here is defined to point at the value of the S attribute, the temporal indexical now at the value of the T attribute, the pro1 (pronoun for first person) indexical at the value of the A attribute, the pro2 (pronoun for second person) indexical at the value of the R attribute, and the pro3 (pronoun for third person) indexical at the value of the 3rd attribute.

The classification of a STAR-0 content as an internal recognition is based on the following pattern, shown as it is matching the content token 6.3.2:

6.3.5 PATTERN CHARACTERIZING 6.3.2 AS INTERNAL RECOGNITION

pattern level	noun: pro1	verb: feel	adj: α	S: L
	fnc: feel	sem: pres ind	mdd: feel	T: D
	prn: K	arg: pro1	prn: K	A: N
		mdr: α		R:
		prn: K		3rd:
				prn: K

where α ε {angry, bold, cold, dumb, excited, furious, greedy, happy, hot, hungry, ...}

content level	noun: pro1	verb: feel	adj: hungry	S:kitchen
	fnc: feel	sem: pres ind	mdd: feel	T: 20080915T155300
	prn: 22	arg: pro1	prn: 22	A: sylvester
		mdr: hungry		R:
		prn: 22		3rd:
				prn: 22

The matching between the pattern and content proplets is based on (i) their having the same attributes in the same order, (ii) the compatibility between

[7] Because a STAR's core attribute A is the third letter in the acronym, A is in the third line of a STAR proplet and the owner value is correspondingly in the third instead of the usual second line (1.5.1).

corresponding variables[8] and constants, and (iii) identity between corresponding constants at the pattern and the content level (1.6.1). The pattern may be used to find all contents (Sect. 14.1) in the agent's A-memory which share the values pro1, feel, and pres ind, and comply with the variable restriction on α.

Internal and external recognition are alike in that (i) they are anchored to a STAR-0 and (ii) the verb is in the present indicative (also called simple present tense). They differ in that the subject of internal recognition is pro1 while the subject of external recognition is non-first-person,[9] as in, for example, You fall into pool, Hector fall into pool, or he fall into pool:

6.3.6 EXTERNAL RECOGNITION You fall into pool. BOUND TO STAR-0

$$\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: fall}\\ \text{prn: 24}\end{bmatrix}\begin{bmatrix}\text{verb: fall}\\ \text{sem: pres ind}\\ \text{arg: pro2}\\ \text{mdr: pool}\\ \text{prn: 24}\end{bmatrix}\begin{bmatrix}\text{noun: pool}\\ \text{sem: }\textit{into}\\ \text{mdd: fall}\\ \text{prn: 24}\end{bmatrix}\begin{bmatrix}\text{S: yard}\\ \text{T: Thursday}\\ \text{A: sylvester}\\ \text{R: hector}\\ \text{3rd:}\\ \text{prn: 24}\end{bmatrix}$$

The pro2 pointer serving as the subject is provided with the R value Hector of the STAR-0. Alternatively, the subject may be specified directly as a third-person nonindexical noun, or as pro3 which relies on a 3rd value in the STAR-0 for interpretation (7.1.3–7.1.5; CLaTR 10.1.5, 10.1.6).

The properties of a content resulting from current recognition may be formalized abstractly as the following pattern:

6.3.7 PATTERN CHARACTERIZING 6.3.6 AS EXTERNAL RECOGNITION

$$\begin{array}{ll}\textit{pattern level} & \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{verb: }\beta\\ \text{sem: pres ind}\\ \text{arg: }\alpha\text{ X}\\ \text{prn: K}\end{bmatrix}\qquad\qquad\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix}\\ & \text{where } \alpha \neq \text{pro1}\\ \textit{content level} & \begin{bmatrix}\text{noun: pro2}\\ \text{fnc: fall}\\ \text{prn: 24}\end{bmatrix}\begin{bmatrix}\text{verb: fall}\\ \text{sem: pres ind}\\ \text{arg: pro2}\\ \text{mdr: pool}\\ \text{prn: 24}\end{bmatrix}\begin{bmatrix}\text{noun: pool}\\ \text{sem: }\textit{into}\\ \text{mdd: fall}\\ \text{prn: 24}\end{bmatrix}\begin{bmatrix}\text{S: yard}\\ \text{T: Thursday}\\ \text{A: sylvester}\\ \text{R: hector}\\ \text{3rd:}\\ \text{prn: 24}\end{bmatrix}\end{array}$$

The classification of the content as a current external recognition by the agent is based on the condition $\alpha \neq$ pro1 and the sem values pres ind.

[8] Mnemonically, the S variable L stands for location, the T variable D for datetime, the A variable N for name, the optional R variable P for partner in discourse, and the optional 3rd variable O for other.

[9] When reporting an ongoing episodic event in English, the speaker often uses the present progressive ("I am watching TV."), but indicative is possible, as in "And Kevin scores!!!". Generic content is usually in the present indicative, but past indicative is possible as well (Sect. 6.6).

An agent may regard the credibility of a content resulting from current (internal or external) recognition as higher than one transmitted by means of language. However, current recognition may be faulty. In natural agents, this is not only shown by phenomena such as phantom pain and optical illusions, but also by the wide divergence of witness reports of the same event, amply documented in official records. Thus current recognition contents may be viewed more like a starting point for reasoning which may include language content.

6.4 Invariance Constraint and Identity in Change

A content word can not be of more than one Semantic kind simultaneously (even in conversion[10]), but the Semantic kind *referent* (1.5.3) occurs as the Content kinds *concept, indexical,* and *name* (1.5.4)

6.4.1 THREE CONTENT KINDS OF THE SEMANTIC KIND REFERENT

$$
\begin{array}{ccc}
\textit{concept} & \textit{indexical} & \textit{name} \\
\begin{bmatrix} \text{sur:} \\ \text{noun: animal} \\ \text{cat: sn} \\ \text{...} \end{bmatrix} &
\begin{bmatrix} \text{sur:} \\ \text{noun: pro2} \\ \text{cat: sp2} \\ \text{...} \end{bmatrix} &
\begin{bmatrix} \text{sur: tom} \\ \text{noun: [person x]} \\ \text{cat: snp} \\ \text{...} \end{bmatrix}
\end{array}
$$

The three Content kinds of the Semantic kind *referent* all have literal use, but only the concepts allow figurative use.

The Semantic kind *property* occurs as the Content kinds (i) *concept* and (ii) *indexical*, but not *name*. The concept proplets may have the cat values adn, adv, adnv,[11] and n-s3′ v (1.5.3):

6.4.2 TWO CONTENT KINDS OF THE SEMANTIC KIND PROPERTY

$$
\begin{array}{ccc}
\textit{concept} & \textit{indexical} & \textit{name} \\
\begin{bmatrix} \text{sur:} \\ \text{adj: great} \\ \text{cat: adn} \\ \text{...} \end{bmatrix} &
\begin{bmatrix} \text{sur:} \\ \text{adj: now} \\ \text{cat: adnv} \\ \text{...} \end{bmatrix} & \\
\begin{bmatrix} \text{sur:} \\ \text{noun: table} \\ \text{cat: adnv} \\ \text{sem: \textit{on} def sg} \\ \text{...} \end{bmatrix} &
\begin{bmatrix} \text{sur:} \\ \text{adj: there} \\ \text{cat: adnv} \\ \text{...} \end{bmatrix} & \\
\begin{bmatrix} \text{sur:} \\ \text{verb: melt} \\ \text{cat: n-s3}' \text{ v} \\ \text{...} \end{bmatrix} & &
\end{array}
$$

[10] In word formation, conversion is the use of a single surface, e.g. work, for more than one Syntactic kind, here noun and verb. In DBS, conversion is treated as a lexical ambiguity.

Of the properties, only the concepts may refer nonliterally.

The Syntactic kind transitive verb with its Semantic kind *relation* exists only as the Content kind *concept*, but not as indexical or name:

6.4.3 One Content kind of the Semantic kind relation

concept	*indexical*	*name*
[sur: verb: steal cat: n-s3′ a′ v ...]		
[sur: verb: give cat: n-s3′ d′ a′ v ...]		

Being concepts, transitive verbs have literal and nonliteral use.

The limitation of figurative use to concepts is further restricted by the following constraint:

6.4.4 Invariance Constraint

> A figurative use and its literal counterpart must be of the same Syntactic and Semantic kind (1.5.3, 1.5.5).

For example, one cannot use a 1-place verb like bark to refer figuratively to a noun/referent like dog unless bark is nominalized, as in the little barker (i.e. by turning the property bark into the referent barker). Similarly for the adj fat, which for figurative use must be turned into a noun, as in the old fatso. Functionally, the constraint helps the hearer to find the literal counterpart of a figurative use by reducing the search space.

The systematic examples of Chap. 9 all satisfy the invariance constraint:

6.4.5 Syntactic-semantic invariance of figurative use

Semantic kind	Syntactic kind	nonliteral use	literal counterpart	
referent	noun	animal	dog	9.1.2
property	prepnoun	on the table	on the orange crate	9.2.1
property	adn	great	greater than average	9.6.3
property	adv	enough	more than enough	9.6.6
property	intransitive verb	melt	disappear	9.5.1
relation	transitive verb	steal	take over	9.4.2

[11] The adnv modifiers may be (i) elementary (fast) or phrasal (in the park), and (ii) adnominal (tree in the park) or adverbial (walk in the park). Phrasal modifiers, also called prepnouns, are derived from a referent by means of a preposition or an affix, depending on the language. Therefore, a prepnoun like in the park refers by means of park, in contradistinction to the other modifiers, e.g. elementary fast, which do not refer.

The Semantic kind *property* has several Syntactic kinds, while each Syntactic kind, e.g. prepnoun, has only of one Semantic kind, i.e. *property*, regardless of whether the Grammatical kind is used literally or figuratively. The other two Semantic kinds, i.e. *referent* and *relation*, each have only a single syntactic counterpart.

As an example of using all three Semantic kinds figuratively consider the following description of a dog contorting itself catching a frisbee in mid air:

6.4.6 Example using all three Semantic kinds figuratively

The animal flew acrobatically towards the disc

The content obeys the invariance constraint 6.4.4: literal dog and figurative animal are both singular nouns, literal jumped and figurative flew are both finite verbs in the indicative past, literal in a spectacular gymnastic feat and figurative acrobatically are both adverbials (one phrasal, the other elementary), and literal frisbee and figurative disc are both singular nouns. For successful communication, the hearer/reader must relate figurative animal to literal dog and figurative disc to literal frisbee. The relation flew and the property acrobatically, in contrast, need not be tied to a particular instance.

In summary, figurative use is restricted to part of a proposition (elementary or phrasal), works only with concepts, and obeys the invariance constraint. Of the Semantic kinds referent, property, and relation, only the referents (6.4.1) use all three Content kinds concept, indexical, and name with the associated Computational mechanisms (1.5.4) of matching, pointing, and baptism. The properties (6.4.2) use only matching and pointing. The relations (6.4.3) are limited to matching.

While the correct decision of whether or not different instances of the same noun have the same *referent* is essential for communication being successful pragmatically, there is no such requirement for the other two Semantic kinds, i.e. *property* and *relation*. Technically, asserting identity between two instances of a property or of a relation would be easy enough by coding the second occurrence as the address of the first, but it would be empirically wrong..

As an example, consider blue chameleon projects (tongue) and red chameleon lays (eggs) referring to the same reptile at different moments:[12]

[12] "Any science that deals with change, whether phylogenetic change, developmental change, or behavioral change, requires entities that can change and yet retain their identity (e.g., Homo sapiens, my cat, or my diet), because only such entities provide historical continuity." Baum (2002, p. 107).

6.4.7 Presence vs. absence of identity in change

$$\begin{bmatrix}\text{noun: chameleon}\\ \text{sem: indef sg}\\ \text{fnc: project}\\ \text{mdr: blue}\\ \text{prn: 13}\end{bmatrix}\begin{bmatrix}\text{adj: blue}\\ \text{mdd: chameleon}\\ \text{prn: 13}\end{bmatrix}\qquad\begin{bmatrix}\text{noun: (chameleon 13)}\\ \text{sem: def sg}\\ \text{fnc: lay}\\ \text{mdr: red}\\ \text{prn: 20}\end{bmatrix}\begin{bmatrix}\text{adj: red}\\ \text{mdd: (chameleon 13)}\\ \text{prn: 20}\end{bmatrix}$$

The identity between the two referents is encoded by the address value (chameleon 13) in proposition 20. That the properties blue and red and the relations project and lay are not identical does not affect the identity between the two occurrences of the referent chameleon.[13]

6.5 Referent and Shadow in Reference and Coreference

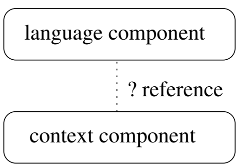
DBS calls the identity relation between two noun proplets *coreference* if they are stored as member proplets in the same token line of A-memory. If the referring proplet is still at the now front, however, the identity relation is called *reference*. Reference automatically becomes coreference when the now front is cleared by moving into fresh memory territory (loom-like clearance), leaving the referent behind in the permanent storage area of the member proplets.

(Co)reference is generalized because the identity relation is coded by using the address of the referred-to item as the core value of the referring item. For this it does not matter whether or not there is an external sur value or an external counterpart: generalized reference works for all four [±surface] and [±external] constellations illustrated in 3.1.1–3.1.3.

The difference between (i) the preliminary conceptual view of reference as a vertical interaction between two separate components in 1.2.2 and the (ii) computational solution as a horizontal relation between noun proplets in the same token line connected by address may be illustrated as follows:

6.5.1 Comparing the naive and the computational solution

Component proposal (preliminary)



Token line solution in A−memory

$$\begin{bmatrix}\text{noun: dog}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: find}\\ \text{mdr: dirty}\\ \text{prn: 16}\end{bmatrix}\cdots\text{(co)reference}\cdots\begin{bmatrix}\text{noun: (dog 16)}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: bark}\\ \text{mdr: clean}\\ \text{prn: 32}\end{bmatrix}$$

The restriction of reference and of coreference to noun proplets in the same token line (token line solution) allows for a precise coding of the compatibility

[13] Another matter are nominalized properties like barker or fatso, which do not serve as modifiers but are regular referents, and similarly for nominalized relations like reader or giver.

conditions: the core value of a referring noun (shadow, copy) at the now front is always an address. The core value of a referred-to noun (referent, original) is never an address. The fnc and mdr values are free (6.4.7).

The following examples 6.5.2 – 6.5.5 show the transition of a proplet from (i) being initial to becoming (ii) a referent to (iii) being coreferent. The example works for nonlanguage and language proplets alike. Consider the arrival of a hitherto unknown referent (original) at the current now front, as when observing an unfamiliar dog:

6.5.2 Initial occurrence of a noun proplet at the now front

	member proplets	*now front*	*owner*
...		$\begin{bmatrix}\text{noun: dog}\\ \text{...}\\ \text{prn: 25}\end{bmatrix}$	dog

Proplets provided by the interface component to the now front, here (dog 25), represent what the agent observes or does in its current reality (+external monitoring).

After subsequent loom-like clearance of the now front by moving it to the right into fresh memory space, the initial occurrence of the *dog* proplet is left behind in what is turning into its final storage position.

6.5.3 Noun proplet after loom-like now front clearance

	member proplets	*now front*	*owner*
...	$\begin{bmatrix}\text{noun: dog}\\ \text{...}\\ \text{prn: 25}\end{bmatrix}$		dog

Later, the now front and the owners may be located even further to the right of the initial referent, depending on the arrival of a few more dog proplets, as when visiting a dog show.

Reference occurs when the original dog is seen again and recognized. This is coded by using the referent's address as the core value of the referring proplet:

6.5.4 Noun proplet referring to initial instance

	member proplets		*now front*	*owner*
...	$\begin{bmatrix}\text{noun: dog}\\ \text{...}\\ \text{prn: 25}\end{bmatrix}$	...	$\begin{bmatrix}\text{noun: (dog 25)}\\ \text{...}\\ \text{prn: 32}\end{bmatrix}$	dog

The referring noun proplet at the current now front is called the *shadow*, in contradistinction to the referred-to original, which is the *referent*. A shadow uses the address, here (dog 25), of the referred to noun proplet as its core value. For simplicity, a shadow proplet may be abbreviated as the address of the referent followed by the prn value of the referring noun containing the address, e.g. ((dog 25) 32) and ((dog 25) 47) in 6.5.5.

As the now front is cleared, the shadow ((dog 25) 32) is left behind as a member proplet. Later, another shadow, e.g. ((dog 25) 47), may be stored at the now front and refer to (dog 25) as well:

6.5.5 REFERENCE AND COREFERENCE

$$
\begin{array}{cccc}
 & \textit{member proplets} & \textit{now front} & \textit{owner} \\
\ldots \begin{bmatrix}\text{noun: dog} \\ \text{...} \\ \text{prn: 25}\end{bmatrix} \ldots & \ldots \begin{bmatrix}\text{noun: (dog 25)} \\ \text{...} \\ \text{prn: 32}\end{bmatrix} \ldots & \begin{bmatrix}\text{noun: (dog 25)} \\ \text{...} \\ \text{prn: 47}\end{bmatrix} & \text{dog}
\end{array}
$$

At his point, the shadow ((dog 25) 47) refers to (dog 25), while the shadow ((dog 25) 32) and the referent (dog 25) are coreferent.

6.6 Generalized Reference

The steps from an initial occurrence (6.5.2) to reference (6.5.4) to coreference (6.5.5) are superimposed by four variants of generalized reference. They result from the distinctions (i) between nouns with and without a sur value and (ii) between the referred-to noun (referent) and the referring noun (shadow). The following example illustrates the four possible constellations of generalized coreference by varying the presence vs. absence of a sur value in the referent or the shadow:

6.6.1 FOUR CONSTELLATIONS OF GENERALIZED COREFERENCE

$$
\begin{array}{ccc}
\textit{member proplets} & \textit{now front} & \textit{owner} \\
\textit{language proplet} & \textit{language proplet} & \\
\ldots \begin{bmatrix}\text{sur: Hund} \\ \text{noun: dog} \\ \text{...} \\ \text{prn: 25}\end{bmatrix} \ldots & \begin{bmatrix}\text{sur: Hund} \\ \text{noun: (dog 25)} \\ \text{...} \\ \text{prn: 32}\end{bmatrix} & \text{dog} \\
\textit{nonlanguage proplet} & \textit{language proplet} & \\
\ldots \begin{bmatrix}\text{sur:} \\ \text{noun: dog} \\ \text{...} \\ \text{prn: 25}\end{bmatrix} \ldots & \begin{bmatrix}\text{sur: Hund} \\ \text{noun: (dog 25)} \\ \text{...} \\ \text{prn: 32}\end{bmatrix} & \text{dog} \\
\textit{language proplet} & \textit{nonlanguage proplet} & \\
\ldots \begin{bmatrix}\text{sur: Hund} \\ \text{noun: dog} \\ \text{...} \\ \text{prn: 25}\end{bmatrix} \ldots & \begin{bmatrix}\text{sur:} \\ \text{noun: (dog 25)} \\ \text{...} \\ \text{prn: 32}\end{bmatrix} & \text{dog}
\end{array}
$$

$$\begin{array}{c} \textit{nonlanguage proplet} \\ \ldots \begin{bmatrix}\text{sur:}\\ \text{noun: dog}\\ \text{...}\\ \text{prn: 25}\end{bmatrix} \ldots \end{array} \qquad\qquad \begin{array}{c} \textit{nonlanguage proplet} \\ \begin{bmatrix}\text{sur:}\\ \text{noun: (dog 25)}\\ \text{...}\\ \text{prn: 32}\end{bmatrix} \end{array} \quad \mathsf{dog}$$

To show the independence of generalized reference from language-dependent surfaces, German Hund is used as the sur value.

Because the processing of (co)reference is run via the semi-universal core value (and not via the language-dependent surface), it is the same in language and nonlanguage cognition. To (i) preserve the prn value of the referent, (ii) provide the shadow with the current prn value, and thus (iii) attach to the current STAR, a shadow by address is a special kind of copy which is derived by an inference like the following:

6.6.2 Inference deriving a noun shadow

$$\begin{array}{lccc}
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} & \begin{bmatrix}\text{noun: }\alpha\\ \text{sem: indef sg}\\ \text{fnc: X}\\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{noun: }(\alpha\text{ K})\\ \text{sem: def sg}\\ \text{fnc:}\\ \text{prn:}\end{bmatrix} \\
 & \alpha\text{ is initial (not an address).} & & \\
 & \Uparrow & & \Downarrow \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} & \begin{bmatrix}\text{sur:}\\ \text{noun: dog}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: bark}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 25}\end{bmatrix} & & \begin{bmatrix}\text{sur:}\\ \text{noun: (dog 25)}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{bmatrix}
\end{array}$$

Potential referents (6.5.2) and shadows (6.5.5) are processed alike.

The simplest use of a shadowed noun is in a new content. For example, if *(dog 25)* is later observed finding a bone, this may be encoded as follows:

6.6.3 Use of a known referent in a new content

$$\begin{bmatrix}\text{sur:}\\ \text{noun: (dog 25)}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 104}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{verb: find}\\ \text{cat: \#n}'\text{ \#a}'\text{ decl}\\ \text{sem: pres ind}\\ \text{arg: (dog 25) bone}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 104}\end{bmatrix} \begin{bmatrix}\text{sur:}\\ \text{noun: bone}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 104}\end{bmatrix} \begin{bmatrix}\text{S: yard}\\ \text{T: Thursday}\\ \text{A: sylvester}\\ \text{R:}\\ \text{3rd:}\\ \text{prn: 104}\end{bmatrix}$$

In this way, the shadow of a referred-to noun (referent) may participate in any new content, referring to the original without touching it.

As an example in which there is no current content at the now front for the shadowed noun to participate in (unlike 6.6.3), consider Mary thinking on Tuesday back to a dog barking on Sunday. Based on the abductive use 3.5.2 of the inference 3.5.1, she reasons that a stranger might have approached.

Her first step is to shadow the stored proposition to the now front:

6.6.4 Shadowing stored proposition to now front

Proposition 25 in memory — *Shadow of proposition 25 at current now front*

$$\begin{bmatrix}\text{sur:}\\ \text{noun: dog}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: bark}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:25}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{verb: bark}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: pres ind}\\ \text{arg: dog}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:25}\end{bmatrix}
\begin{bmatrix}\text{S: yard}\\ \text{T: Sunday}\\ \text{A: mary}\\ \text{R:}\\ \text{3rd:}\\ \text{prn:25}\end{bmatrix}
\Rightarrow
\begin{bmatrix}\text{sur:}\\ \text{noun: (dog 25)}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: bark}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 56}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{verb: bark}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: (dog 25)}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 56}\end{bmatrix}
\begin{bmatrix}\text{S: study}\\ \text{T: Tuesday}\\ \text{A: mary}\\ \text{R:}\\ \text{3rd:}\\ \text{prn: 56}\end{bmatrix}$$

The verb's tense is changed by the inference 7.2.4. The noun shadow (dog 25) in the consequent occurs as the core value of the *dog* proplet and as the continuation value of the bark proplet. This kind of interconnection between a shadowed referent and the other proplets in a content obviates the need to create shadows not only for referents, but also for modifiers and relations.[14]

In a second step, the inference 3.5.1/3.5.2 takes the shadowed proposition as input to the consequent (abductive use) and applies as follows:

6.6.5 Using shadowed consequent for abductive use

$$\begin{array}{lcccc}
\textit{pattern level} &
\begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{noun: dog}\\ \text{fnc: bark}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: bark}\\ \text{arg: fido}\\ \text{prn: K}\end{bmatrix} \\
 & \Downarrow & & \Uparrow & \\
\textit{content level} &
\begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: 57}\end{bmatrix} &
\begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: 57}\end{bmatrix} &
\begin{bmatrix}\text{noun: (dog 25)}\\ \text{fnc: bark}\\ \text{prn: 56}\end{bmatrix} &
\begin{bmatrix}\text{verb: bark}\\ \text{arg: (dog 25)}\\ \text{prn: 56}\end{bmatrix}
\end{array}$$

Accordingly, Mary thinks on Tuesday that a stranger might have approached on Sunday. The original content is accessible via the address (dog 25) in the input. The incremented prn value of the antecedent follows the general principle that the prn value of an output must be higher than that of the input.

Finally consider The earth is flat as a content in memory and The earth is not flat as a belated correction at the now front (shown in horizontal format):

[14] In other words, for the purpose of enabling content in memory to participate in content at the now front, it is sufficient to shadow only the referents; properties (e.g. the bark value in 6.6.3) and relations are used directly (not by address) and rely pragmatically on their semantic connections to shadowed referents.

6.6.6 Correcting stored content

stored content		*correction at now front*	
noun: earth	verb: is_flat	noun: (earth 43)	verb: is_flat
cat: snp	cat: #ns3′ decl	cat: snp	cat: #ns3′ decl
sem: def sg	sem: pres ind	sem: def sg	sem: pres ind *not*
fnc: is_flat	arg: earth	fnc: is_flat	arg: (earth 43)
prn: 43	prn: 43	prn: 58	prn: 58

This is a correction[15] because it is the most recent (rightmost) version of a content in A-memory which represents the agent's current opinion.[16]

In summary, shadowing stored memory content to the now front is (i) for participating in current content (6.6.3), (ii) as input to inferencing (6.6.5), and (iii) for correction (6.6.6). Representing a shadow as an address maintains the fundamental principle that content stored in A-memory may never be touched (though read and copied). It also avoids the unbounded multiplication of copies and the concomitant need to assert their identity (as in coindexing).

Synopsis: The Semantic-Pragmatic Distinction in DBS

In DBS, the semantic-pragmatic distinction applies to content. A content is defined as a set of order-free proplets connected by the semantic relations of structure coded by address. Contents attached to the agent's onboard orientation system are tokens and belong to the pragmatics, while contents without such a connection are types and belong to the semantics.

The relevant parameter values of the agent's on-board orientation system are continuously coded in the form of STAR proplets which specify the agent (pro1) and monitor the agent's moment by moment location in time and space as well as possible relations to other agents or objects. A content and its STAR are connected by a shared prn value. For indexicals in the content, the STAR proplet provides the values to point at.

Pragmatically, STAR proplets are differentiated into (i) STAR-0 proplets of origin, (ii) STAR-1 proplets coding the perspective of the speaker, and (iii) STAR-2 proplets coding the perspective of the hearer. In nonliteral language use, the hearer's interpretation of the speaker's surface must revert the content into the literal original in order for communication to succeed.

---

15 In a correction, the initial referent may also be indefinite, as in A whale is a mammal (natural kind, generic).

16 The ability to change one's mind in the face of evidence has been called a criterion of personhood (Lance and White 2007). There is, however, evidence that the ability is rather rare.

7. STAR-1 Content: Speaker's Perspective

Meaning$_2$ (6.1.1) aspects rely on the STAR of a content token and constitute the pragmatics of language cognition. For example, converting the STAR-0 content I see you into a STAR-1 content provides such options as I see/saw you, I see/saw him, or I see/saw Hector. The choice between them is determined by the speaker's utterance situation and as such obligatory.

In addition there are optional STAR-0 STAR-1 conversions. For example, communicative purpose, e.g. politeness, may motivate a speaker to map a literal STAR-0 content of origin, for example Pass the salt!, into a nonliteral STAR-1 content of production, here Could you pass the salt?

Obligatory STAR-0 STAR-1 adaptations are presented in Sects. 7.1 and 7.2, optional syntactic mood adaptations in Sects. 7.3–7.5, and the STAR-1 adaptation for repeated hearsay in Sect. 7.6. For correct understanding, the intended hearer maps the content of a nonliteral STAR-1 surface back into the literal content, but adjusted to the hear mode perspective (STAR-2 content, Chap. 8).

7.1 From STAR-0 Content to STAR-1 Perspective

A current STAR-0 content immediately and irrevocably recedes into the past. Looking back from the current STAR-0 onto a past STAR-0 content in memory requires the derivation of a STAR-1 content which codes the agent's perspective. Let us use the following STAR-0 content as the starting point:

7.1.1 STAR-0 CONTENT OF ORIGIN: I see you.

			STAR-0 proplet of origin
sur:	sur:	sur:	S: yard
noun: pro1	verb: see	noun: pro2	T: Thursday[1]
cat: s1	cat: #n-s3′ #a′ decl	cat: sp2	A: sylvester
sem: sg	sem: pres ind	sem:	R: hector
fnc: see	arg: pro1 pro2	fnc: see	3rd:
...	...	...	prn: 12
prn: 12	prn: 12	prn: 12	

[1] Using weekdays as T values is crude, but sufficient for the following examples.

By definition, a STAR-0 content is without language and the verb's sem value is pres ind. The pro1 and the pro2 indexicals in the content point at the A value sylvester and the R value hector, respectively, of the STAR-0 proplet.

Depending on the utterance situation, the speaker may derive STAR-1 variants which differ semantically but are pragmatically equivalent to the STAR-0 content 7.1.1.[2]

7.1.2 Pragmatically equivalent variants of STAR-0 content

STAR-1 A: Sylvester remembers the content 7.1.1 without speaking.
STAR-1 B: Sylvester tells Hector that he saw him.
STAR-1 C: Sylvester tells Speedy that he saw "him," referring to Hector.
STAR-1 D: Sylvester tells Speedy that he saw Hector.
STAR-1 E: Sylvester tells Speedy that he saw Hector in the yard.
STAR-1 F: Sylvester tells Speedy that he saw Hector on Thursday in the yard.

In variants A-D, the semantic differences in the contents are compensated pragmatically by varying STAR-1 values. In variants E and F, past STAR-0 values which were overwritten by the current ones are preserved by writing them into the content.

7.1.3 STAR-1 content A: I saw hector. (nonlanguage)

			STAR-1 proplet of production
sur:	sur:	sur: hector	S: veranda
noun: pro1	verb: see	noun: [dog x]	T: Sunday
cat: s1	cat: #n-s3′ #a′ decl	cat: nm m sg	A: sylvester
sem: sg	sem: past ind	sem:	R:
fnc: see	arg: pro1 hector	fnc: see	3rd:
...	...	...	prn: 52
prn: 52	prn: 52	prn: 52	

Compared to 7.1.1, the S- and T-values changed from the yard to veranda and from Thursday to Sunday, respectively. The 3rd-value stays zero because there is no partner of discourse. In the content, the verb's feature [sem: pres ind] is changed to [sem: past ind] and pro2 is changed to Hector with a concomitant deletion of the R-value.

In variant B, sylvester and hector are in contact, and sylvester chooses to tell hector the STAR-0 content 7.1.1. As the addressee, the grammatical object is represented as the indexical pro2 in the content, pointing at the R-value hector in the STAR-1:

[2] The hearer's pragmatically equivalent STAR-2 counterparts are listed in 8.1.2.

7.1.4 STAR-1 CONTENT B: I saw you.

			STAR-1 proplet of production
sur:	sur:	sur:	S: veranda
noun: pro1	verb: see	noun: pro2	T: Sunday
cat: s1	cat: #n-s3′ #a′ decl	cat: sp2	A: sylvester
sem: sg	sem: past ind	sem:	R: hector
fnc: see	arg: pro1 pro2	fnc: see	3rd:
...	...	...	prn: 52
prn: 52	prn: 52	prn: 52	

Compared to the STAR-0 content, the S- and T- values may have changed, the A-value must stay the same, the R-value hector is for pro2 to point at, and the 3rd-value is zero.

In variant C, Sylvester talks to Speedy about Hector using a pro3 indexical:

7.1.5 STAR-1 CONTENT C: I saw him.

			STAR-1 proplet of production
sur:	sur:	sur:	S: veranda
noun: pro1	verb: see	noun: pro3	T: Sunday
cat: s1	cat: #n-s3′ #a′ decl	cat: obq	A: sylvester
sem: sg	sem: past ind	sem: sg m	R: speedy
fnc: see	arg: pro1 pro3	fnc: see	3rd: hector
...	...	...	prn: 52
prn: 52	prn: 52	prn: 52	

In the STAR-1 proplet, the value of the R-attribute is speedy and of the 3rd-attribute hector. In the content, the grammatical object changes from [noun: pro2] to [noun: pro3] and from [cat: sp2] to [cat: obq] (oblique).

Variant D is like variant C except that sylvester refers to the grammatical object by name, and not by pronoun:

7.1.6 STAR-1 CONTENT D:I saw Hector.

			STAR-1 proplet of production
sur:	sur:	sur: hector	S: veranda
noun: pro1	verb: see	noun: [dog x]	T: Sunday
cat: s1	cat: #n-s3′ #a′ decl	cat: snp	A: sylvester
sem: sg	sem: past ind	sem: nm m sg	R: speedy
fnc: see	arg: pro1 [dog x]	fnc: see	3rd:
...	...	...	prn: 52
prn: 52	prn: 52	prn: 52	

Without a pro3 indexical in the content, there is no need to write hector into the 3rd slot of the STAR-1.

The variants E and F preserve the STAR-0 proplet's S value yard and/or T value Thursday, which were systematically overwritten with those of the current STAR-0 proplet (7.1.3–7.1.6). The method of preserving the information is to add it to the content.

In variant E, it is the T value Thursday which is being preserved:

7.1.7 STAR-1 CONTENT E: I saw Hector on Thursday.

				STAR-1 proplet
sur:	sur:	sur: hector	sur:	S: veranda
noun: pro1	verb: see	noun: [dog x]	noun: Thursday	T: Sunday
cat: s1	cat: #n-s3′ #a′ decl	cat: snp	cat: snp	A: sylvester
sem: sg	sem: past ind	sem: nm m sg	sem: *on*	R: speedy
fnc: see	arg: pro1 [dog x]	fnc: see	mdd: see	3rd:
...	mdr: Thursday	...	...	prn: 52
prn: 52	...	prn: 52	prn: 52	
	prn: 52			

The STAR-1 proplets of this and the next content are the same as in 7.1.6.

In variant F, the speaker preserves the STAR-0's S value yard by adding it as a prepnoun proplet into the STAR-1 content 7.1.7:

7.1.8 VARIANT F: STAR-1 CNT. I saw Hector on Thursday in the yard.

					STAR-1 proplet
sur:	sur:	sur: hector	sur:	sur:	S: veranda
noun: pro1	verb: see	noun: [dog x]	noun: Thursday	noun: yard	T: Sunday
cat: s1	cat: #n-s3′ #a′ decl	cat: snp	cat: snp	cat: snp	A: sylvester
sem: sg	sem: past ind	sem: nm m sg	sem: *on*	sem: *in*	R: speedy
fnc: see	arg: pro1 [dog x]	fnc: see	mdd: see	mdd: see	3rd:
...	mdr: Thursday yard	...	...	...	prn: 52
prn: 52	...	prn: 52	prn: 52	prn: 52	
	prn: 52				

Because adverbial modifiers are essentially order-free in English (as in many other natural languages), there are quite a few variants and paraphrases. The examples in the following list may all be formally explicated like 7.1.3–7.1.8:

7.1.9 STAR-1 SURFACE VARIANTS OF THE STAR-0 CONTENT 7.1.1

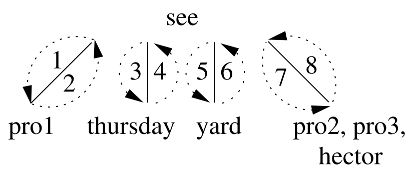
I saw you
I saw you in the yard
I saw you on Thursday
I saw you in the yard on Thursday
I saw you on Thursday in the yard
On Thursday I saw you
In the yard I saw you
On Thursday I saw you in the yard
...
I saw him
I saw him in the yard
I saw him on Thursday
I saw him in the yard on Thursday
I saw him on Thursday in the yard
...
I saw Hector
I saw Hector in the yard
I saw Hector on Thursday

I saw Hector in the yard on Thursday
I saw Hector on Thursday in the yard
...

Pragmatically, these different STAR-1 surfaces represent the STAR-0 content 7.1.1 equivalently. The grammatical object may be written into the R slot (7.1.4), the 3rd slot (7.1.5), or the content (7.1.6), the adverbial modifiers may be either omitted (7.1.3–7.1.6) or written into the content (7.1.7, 7.1.8), and the tense may be pres ind or past ind, depending on the moment of utterance.

The content 7.1.1 common to the variants of 7.1.9 may be shown as the following semantic relations graph:

7.1.10 SEMANTIC RELATIONS GRAPH UNDERLYING THE 7.1.6 VARIANTS



Given that the proplets of a content are order-free, the surface position of the adverbial modifiers is speak rather than think mode specific and as such an aspect of the language-dependent navigation order. Whether or not the traversal of the modifiers results in surfaces is also a matter of the speak mode, namely of whether the lexicalization rules (2.6.2, 2.6.3) are switched on or off.

Outside the laboratory set-up, the surface variants depend not only on the utterance situation, but also on the speaker's state of mind. For example, whether or not the S and/or T information of the STAR-0 proplet of origin is omitted (7.1.3–7.1.6) or added into the STAR-1 content of production (7.1.7, 7.1.8) depends on whether or not it is (a) important to the speaker or (b) considered redundant, i.e. attributed by the speaker to the hearer as shared knowledge.

7.2 Indexical STAR-0 STAR-1 Inference

In the medium of written language, the moment and location of (i) a content's origin (STAR-0), (ii) its language production in the speak mode (STAR-1), and (iii) its language interpretation in the hear mode (STAR-2) may be arbitrarily far apart in time and space. For example, Hector fall into pool may originate by direct observation on Thursday in the yard (STAR-0 content), be produced in writing on Sunday on the veranda (STAR-1 content), and be interpreted on Wednesday by the intended hearer (reader) in another town (STAR-2 content).

Thereby, the STAR-1 content is the pivot of communication: it is produced by the speaker for the intended hearer to interpret. While in spoken face-to-face communication the STAR-1 is given by the utterance situation, this is usually not the case in written language. To ensure successful communication, personal letters specify the values of the STAR-1 proplet by convention.[3]

For adjusting the spatio-temporal indexicality of a STAR-0 STAR-1 content transition, the speak mode uses an inference which takes (i) the past STAR-0 content and the (ii) current STAR-0 proplet as input and produces a current STAR-1 content as output. For illustration, let us use the past STAR-0 content 7.1.1 as the first input item, repeated for convenience:

7.2.1 First input item to indexical STAR-0 STAR-1 inference

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: see}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{verb: see}\\ \text{sem: pres ind}\\ \text{arg: pro1 pro2}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: see}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{S: yard}\\ \text{T: Thursday}\\ \text{A: sylvester}\\ \text{R: hector}\\ \text{3rd:}\\ \text{prn: 12}\end{bmatrix}$$

Let us assume that the speaker's current STAR-0 is defined as follows:

7.2.2 Second input item to indexical STAR-0 STAR-1

$$\begin{bmatrix}\text{S: study}\\ \text{T: Sunday}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52}\end{bmatrix}$$

The intended output of the inference for these two input items is as follows:

7.2.3 Output of the indexical STAR-0 STAR-1 inference 7.2.4

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: see}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{verb: see}\\ \text{sem: past ind}\\ \text{arg: pro1 pro3}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{noun: pro3}\\ \text{fnc: see}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{S: study}\\ \text{T: Sunday}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd: hector}\\ \text{prn: 52}\end{bmatrix}$$

The reason for changing the S-, T-, R-, 3rd-, and prn values of the past STAR-0 proplet to those of the current STAR-0 proplet is the inevitable progression of time (T- and prn value), and the concomitant changing of spatial landmarks (S-value) and the partner of discourse (R- and 3rd-values).

Using the past STAR-0 content 7.1.1/7.2.1, and the current STAR-0 proplet 7.2.2 as input, the following inference derives a STAR-1 content of production, corresponding to pro1 saw pro3 as told by Sylvester to Speedy, with pro3 pointing at hector, which is being demoted from the STAR-0 proplet's R value (7.2.1) to the STAR-1 proplet's 3rd value:

7.2.4 Indexical STAR-0 STAR-1 inference

$$
\begin{array}{llllll}
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} &
\left[\begin{array}{l}\text{noun: pro1}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{array}\right]
\left[\begin{array}{l}\text{verb: }\alpha\\ \text{sem: pres ind}\\ \text{arg: pro1 }\delta\\ \text{prn: K}\end{array}\right]
\left[\begin{array}{l}\text{noun: }\delta\\ \text{fnc: }\alpha\\ \text{prn: K}\end{array}\right]
\left[\begin{array}{l}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{array}\right]
\left[\begin{array}{l}\text{S: L}'\\ \text{T: D}'\\ \text{A: N}\\ \text{R: P'}\\ \text{3rd:}\\ \text{prn: K}'\end{array}\right] & \Rightarrow \\
& \Uparrow \qquad\qquad \Uparrow \qquad\qquad \Uparrow \qquad\qquad \Uparrow \qquad\qquad \Uparrow & \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\left[\begin{array}{l}\text{noun: pro1}\\ \text{fnc: see}\\ \text{prn: 12}\end{array}\right]
\left[\begin{array}{l}\text{verb: see}\\ \text{sem: pres ind}\\ \text{arg: pro1 pro2}\\ \text{prn: 12}\end{array}\right]
\left[\begin{array}{l}\text{noun: pro2}\\ \text{fnc: see}\\ \text{prn: 12}\end{array}\right]
\left[\begin{array}{l}\text{S: yard}\\ \text{T: Thursday}\\ \text{A: sylvester}\\ \text{R: hector}\\ \text{3rd:}\\ \text{prn: 12}\end{array}\right]
\left[\begin{array}{l}\text{S: study}\\ \text{T: Sunday}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52}\end{array}\right] &
\end{array}
$$

$$
\begin{array}{ll}
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} &
\left[\begin{array}{l}\text{noun: pro1}\\ \text{fnc: }\alpha\\ \text{prn: K}'\end{array}\right]
\left[\begin{array}{l}\text{verb: }\alpha\\ \text{sem: }\gamma\\ \text{arg: pro1 P}\\ \text{prn: K}'\end{array}\right]
\left[\begin{array}{l}\text{noun: }\delta'\\ \text{fnc: }\alpha\\ \text{prn: K}'\end{array}\right]
\left[\begin{array}{l}\text{S: L}'\\ \text{T: D}'\\ \text{A: N}\\ \text{R: P'}\\ \text{3rd: P}\\ \text{prn: K}'\end{array}\right] \\
& \text{(1) If D}' > \text{D, then } \gamma = \text{past; if D}' = \text{D, then } \gamma = \text{pres.} \\
& \text{(2) If } \delta = \text{pro2, then } \delta' = \text{pro3; if P = P}'\text{, then } \delta' = \text{pro2; otherwise } \delta' = \text{P.} \\
& \Downarrow \qquad\qquad \Downarrow \qquad\qquad \Downarrow \qquad\qquad \Downarrow \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\left[\begin{array}{l}\text{noun: pro1}\\ \text{fnc: see}\\ \text{prn: 52}\end{array}\right]
\left[\begin{array}{l}\text{verb: see}\\ \text{sem: past ind}\\ \text{arg: pro1 [dog x]}\\ \text{prn: 52}\end{array}\right]
\left[\begin{array}{l}\text{noun: pro3}\\ \text{fnc: see}\\ \text{prn: 52}\end{array}\right]
\left[\begin{array}{l}\text{S: study}\\ \text{T: Sunday}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd: hector}\\ \text{prn: 52}\end{array}\right]
\end{array}
$$

The inference applies only in the speak mode and only deductively. It adjusts the tense in the content from pres to past and preserves the old R-value hector as the new 3rd value. The $'$ marker[4] is used here to indicate a possible alternative value; for example, L and L$'$ may be different locations. If D = D$'$, as in life reporting, the sem value of the finite verb is pres ind; if D < D$'$, in contrast, the sem value is past ind.

---

[3] In a personal letter, the convention is as follows: the location (S value) and the date (T value) of production are written at the top right of the first page, the author (A value) at the end in the farewell greeting, and the intended recipient (R value) in the opening greeting. In addition, the official name and the address of the intended recipient is written on the front side of the envelope, while those of the author (sender) is written on the back side (FoCL Sect. 5.3; CLaTR Sect. 11.1).

[4] This use of $'$ is distinct from its use to indicate a valency position in a cat slot, e.g. a$'$.

7.3 Adapting STAR-0 Imperative into STAR-1 Interrogative

The STAR-0 STAR-1 adaptations described in Sects. 7.1 and 7.2 are literal in that they update spatio-temporal indexicality as an obligatory aspect of the speaker's STAR-1 perspective. Let us turn next to optional STAR-1 adaptations of STAR-0 contents, beginning with the syntactic[5] mood adaptations.[6]

In the following example, an abrupt command is softened into a polite request by a DBS inference which changes an imperative into an interrogative:

7.3.1 Input: imperative, output: interrogative

STAR-0 content input: Pass the salt!
STAR-1 content output: Could you pass the salt?

The inference is called IMP-INT (imperative-interrogative). It is applied deductively in the speak and abductively in the hear mode.

7.3.2 IMP-INT inference for STAR-0 STAR-1 mood adaptation

$$
\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: n-s3}'\ \#\text{a}'\ \text{impv}\\ \text{arg: pro2 }\beta\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: }\beta\\ \text{cat: def sg}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix}
\Rightarrow
\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: }\#\text{n}'\ \#\text{a}'\ \text{interrog}\\ \text{sem: sbj } \textit{can}\\ \text{arg: pro2 (}\beta\text{ K)}\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: }\alpha\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{noun: }\beta\\ \text{cat: def sg}\\ \text{fnc: }\alpha\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix}
$$

If β ε {salt, pepper, potatoes}, then α = pass. If β ε {window, door, hatch}, then α ε {open, close}, etc.

$\Uparrow \quad \Uparrow \quad \Uparrow \quad \Downarrow \quad \Downarrow \quad \Downarrow \quad \Downarrow$

$$
\begin{bmatrix}\text{verb: pass}\\ \text{cat: n-s3}'\ \#\text{a}'\ \text{impv}\\ \text{arg: pro2 salt}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{noun: salt}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{verb: pass}\\ \text{cat: }\#\text{n}'\ \#\text{a}'\ \text{interrog}\\ \text{sem: sbj } \textit{can}\\ \text{arg: pro2 (salt 52)}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{noun: salt}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52}\end{bmatrix}
$$

The antecedent's STAR-0 pattern and the consequent's STAR-1 pattern differ in their prn variables and in the syntactic mood value impv vs. interrog of the verbs' cat attribute. Using the restrictions on α and β, the inference produces such adapted speak mode contents as Could you pass the salt/pepper/potatoes/...?, Could you open/close the window/door/hatch/...?, etc.

The semantic relations resulting from applying the IMP-INT inference may be represented as the following graphical analysis:

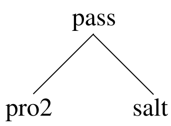
---

[5] The syntactic moods, e.g., interrogative vs. imperative, are distinct from the verbal moods, e.g. indicative vs. subjunctive.

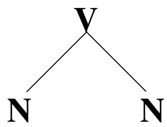
[6] Syntactic mood adaptations occur in many European languages, but are not universal. The alternative in Korean, for example, is the use of two morphological systems, one for honor and one for mood, which are agglutinated to the verbal stem. Thanks to Prof. Kiyong Lee for his help in this matter.

7.3.3 CANONICAL DBS GRAPH ANALYSIS OF Could you pass the salt?

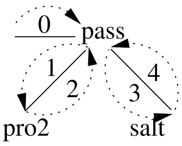
(i) SRG (semantic relations graph)



(ii) signature



(iii) NAG (numbered arcs graph)



(iv a) surface realization English

0	1	2	3	4
Could	you	pass	the_salt	?
−−V	V/N	N/V	V\N	N\V

(iv b) surface realization German

0	1	2	3	4
Könnten	Sie		das_Salz	reichen_?
−−V	V/N	N/V	V\N	N\V

Let us use the German surface *(iv b)* produced in the speak mode as input to the associated hear mode derivation. This is the step prior to the hearer's reversion of an assumed nonliteral to the corresponding literal content (Sect. 8.3).

In DBS, automatic recognition of the next word and its syntactic-semantic integration into the current sentence start are incrementally intertwined, i.e. a new next word is looked up only after the current next word has been integrated into the current sentence start (time-linearity):

7.3.4 CANONICAL HEAR MODE DERIVATION OF GERMAN SURFACE

Könnten	Sie	das	Salz	reichen	?
lexical lookup					
sur: Könnten	sur: Sie	sur: das	sur: Salz	sur: reichen	sur: ?
verb: v_1	noun: pro2	noun: n_1	noun: salt	verb: pass	verb: v_2
cat: ns2' can' v	cat: sp2	cat: −mf' sna	cat: −mf	cat: np13' a' v	cat: vi' interrog
sem: sbj *can*	sem:	sem: def sg	sem:	sem: pres	sem: pres
arg:	fnc:	fnc:	fnc:	arg:	arg:
mdr:	mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:	pc:
prn:	prn:	prn:	prn:	prn:	prn:

syntactic−semantic parsing:

	sur: Könnten	sur: Sie	
1	verb: v_1	noun: pro2	*cross−copying*
	cat: ns2' can' v	cat: sp2	
	sem: sbj *can*	sem:	
	arg:	fnc:	
	mdr:	mdr:	
	nc:	nc:	
	pc:	pc:	
	prn: 24	prn:	

2

sur:	sur:	sur: das	*cross−copying*
verb: v_1	noun: pro2	noun: n_1	
cat: #ns2' can' vi	cat: sp2	cat: −mf' sna	
sem: sbj *can*	sem:	sem: def sg	
arg: pro2	fnc: v_1	fnc:	
mdr:	mdr:	mdr:	
nc:	nc:	nc:	
pc:	pc:	pc:	
prn: 24	prn: 24	prn:	

3

sur:	sur:	sur:	sur: Salz	*absorption with simultaneous substitution*
verb: v_1	noun: pro2	noun: n_1	noun: salt	
cat: #ns2' #can' vi	cat: sp2	cat: −mf' sna	cat: −mf	
sem: sbj *can*	sem:	sem: def sg	sem:	
arg: pro2 n_1	fnc: v_1	fnc: v_1	fnc:	
mdr:	mdr:	mdr:	mdr:	
nc:	nc:	nc:	nc:	
pc:	pc:	pc:	pc:	
prn: 24	prn: 24	prn: 24	prn:	

4

sur:	sur:	sur:	sur: reichen	*absorption with simultaneous substitution*
verb: v_1	noun: pro2	noun: salt	verb: pass	
cat: #ns2' #can' vi	cat: sp2	cat: sna	cat: np13' a' v	
sem: sbj *can*	sem:	sem: def sg	sem: pres	
arg: pro2 salt	fnc: v_1	fnc: v_1	arg:	
mdr:	mdr:	mdr:	mdr:	
nc:	nc:	nc:	nc:	
pc:	pc:	pc:	pc:	
prn: 24	prn: 24	prn: 24	prn:	

5

sur:	sur:	sur:	sur: ?	*absorption*
verb: pass	noun: pro2	noun: salt	verb: v_2	
cat: #ns2' #a' vi	cat: sp2	cat: sna	cat: vi' interrog	
sem: sbj *can*	sem:	sem: def sg	sem: pres	
arg: pro2 salt	fnc: pass	fnc: pass	arg:	
mdr:	mdr:	mdr:	mdr:	
nc:	nc:	nc:	nc:	
pc:	pc:	pc:	pc:	
prn: 24	prn: 24	prn: 24	prn:	

result:

sur:	sur:	sur:
verb: pass	noun: pro2	noun: salt
cat: #ns2' #a' interrog	cat: sp2	cat: sna
sem: sbj *can*	sem:	sem: def sg
arg: pro2 salt	fnc: pass	fnc: pass
mdr:	mdr:	mdr:
nc:	nc:	nc:
pc:	pc:	pc:
prn: 24	prn: 24	prn: 24

This derivation of German Könnten Sie[7] das Salz reichen? corresponds to English Could you pass the salt?[8] An analogous 'construction' (Fillmore 1988) is adapting Have some more! into Would you have some more?

[7] In German, the pronoun Sie is the polite third person plural in the nominative and accusative, here fudged as pro2 for simplicity. Also, the Abtönungspartikel (modal particle) bitte/please is omitted.

[8] Replacing the valency position can′ of the modal verb with the a′ value from the nonfinite main verb

7.4 Adapting STAR-0 Interrogative into STAR-1 Declarative

A second syntactic mood adaptation is the optional strengthening of a STAR-0 Yes/No interrogative into a STAR-1 declarative, as in the following example:

7.4.1 INPUT: INTERROGATIVE, OUTPUT: DECLARATIVE

STAR-0 content input: Did John pass the salt?
STAR-1 content output: I am asking you whether John passed the salt.

The higher clause in the STAR-1 content is called an *explicit performative* in Speech Act Theory (Austin [1955] 1962).

The inference introducing this syntactic mood adaption is called INT-DECL (interrogative-declarative). Like 13.1.3, the inference is applied deductively in the speak mode and abductively in the hear mode.

7.4.2 INT-DECL INFERENCE FOR STAR-0 STAR-1 MOOD ADAPTATION

$$
\begin{array}{llllll}
\textit{pattern level} &
\begin{bmatrix}\text{verb: }\gamma\\ \text{sem: interrog}\\ \text{arg: }\alpha\ \beta\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\gamma\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\gamma\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix}
\Rightarrow
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: ask}\\ \text{prn: K+1}\end{bmatrix}
\\
& \Uparrow \qquad \Uparrow \qquad \Uparrow \qquad \Uparrow \qquad \Downarrow
\\
\textit{content level} &
\begin{bmatrix}\text{verb: pass}\\ \text{arg: john salt}\\ \text{sem: interrog past } \textit{do}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{noun: john}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{noun: salt}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52}\end{bmatrix}
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: ask}\\ \text{prn: 53}\end{bmatrix}
\end{array}
$$

$$
\begin{array}{ll}
\textit{pattern level} &
\begin{bmatrix}\text{verb: ask}\\ \text{sem: pres prog}\\ \text{arg: pro1 pro2 }(\gamma\text{ K})\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: ask}\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{verb: }(\gamma\text{ K})\\ \text{sem: }\textit{whether}\\ \text{fnc: (ask K+1)}\\ \text{arg: }(\alpha\text{ K})\ (\beta\text{ K})\\ \text{prn: K+1+1}\end{bmatrix}
\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\gamma\\ \text{prn: K+1+1}\end{bmatrix}
\begin{bmatrix}\text{noun: }(\beta\text{ K})\\ \text{fnc: }\gamma\\ \text{prn: K+1+1}\end{bmatrix}
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K+1+1}\end{bmatrix}
\\
& \Downarrow \qquad \Downarrow \qquad \Downarrow \qquad \Downarrow \qquad \Downarrow \qquad \Downarrow
\\
\textit{content level} &
\begin{bmatrix}\text{verb: ask}\\ \text{sem: pres prog}\\ \text{arg: pro1 pro2 (pass 52)}\\ \text{cat: \#s1}'\text{ \#a}'\text{ decl}\\ \text{prn: 53}\end{bmatrix}
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: ask}\\ \text{prn: 53}\end{bmatrix}
\begin{bmatrix}\text{verb: (pass 52)}\\ \text{sem: }\textit{whether}\\ \text{fnc: (ask 52)}\\ \text{arg: john (salt 52)}\\ \text{prn: 54}\end{bmatrix}
\begin{bmatrix}\text{noun: john}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn:54}\end{bmatrix}
\begin{bmatrix}\text{noun: (salt 52)}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 54}\end{bmatrix}
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52-54}\end{bmatrix}
\end{array}
$$

---

in line 4 (shown in line 5) is needed for verb agreement in German. For example, reichen takes an accusative, but helfen takes a dative, as in Könntest Du dem Kind helfen?. This is buffered by the can′ valency position until the arrival of the main verb provides the information needed to prevent ungrammatical, e.g., *Könntest Du das Kind helfen? or *Könntest Du dem Salz reichen?. The actual fillers are listed in the arg slot of the finite verb proplet resulting from absorption (line 4), here [arg: pro2 salt].

At the pattern level, there are no variables in the performative clause I am asking you, except for the prn values. At the content level, the remainder of the consequent resembles the STAR-0 input content. Syntactically, the consequent is a standard object clause construction (TExer Sect. 2.6).

7.5 Adapting STAR-0 Imperative into STAR-1 Declarative

A third syntactic mood adaptation is the optional strengthening of a STAR-0 imperative into a STAR-1 declarative, as in the following example:

7.5.1 INPUT: IMPERATIVE, OUTPUT: DECLARATIVE

STAR-0 input content: Pass the salt!
STAR-1 output content: I am ordering you to pass the salt.

The associated inference is called IMP-DECL (imperative-declarative). Like 13.1.3 and 7.4.2, the inference applies deductively in the speak mode and abductively in the hear mode. (For related infinitive constructions see CLaTR Sect. 15.6; TExer Sect. 4.4.)

7.5.2 IMP-DECL INFERENCE FOR STAR-0 STAR-1 MOOD ADAPTATION

$$\textit{pattern level}\quad \begin{bmatrix}\text{verb: }\beta\\ \text{cat: n-s3}'\ \#\text{a}'\text{ impv}\\ \text{sem: pres ind}\\ \text{arg: pro2 }\alpha\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: }\beta\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: order}\\ \text{prn: K+1}\end{bmatrix} \begin{bmatrix}\text{verb: order}\\ \text{cat: n}'\text{ a}'\text{ inf}'\text{ decl}\\ \text{sem: pres prog}\\ \text{arg: pro1 pro2 }\beta\\ \text{prn: K+1}\end{bmatrix} \begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: order}\\ \text{prn: K+1}\end{bmatrix}$$

If $\alpha\ \varepsilon$ {salt, pepper, potatoes}, then β = pass. If $\alpha\ \varepsilon$ {window, door}, then $\beta\ \varepsilon$ {open, close}, etc.

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

$$\textit{content level}\quad \begin{bmatrix}\text{verb: pass}\\ \text{cat: n-s3}'\ \#\text{a}'\text{ impv}\\ \text{sem: pres ind}\\ \text{arg: pro2 salt}\\ \text{prn: 52}\end{bmatrix} \begin{bmatrix}\text{noun: salt}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix} \begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52}\end{bmatrix} \quad \begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: order}\\ \text{prn: 53}\end{bmatrix} \begin{bmatrix}\text{verb: order}\\ \text{cat: n}'\text{ a}'\text{ inf}'\text{ decl}\\ \text{sem: pres prog}\\ \text{arg: pro1 pro2 pass}\\ \text{prn: 53}\end{bmatrix} \begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: order}\\ \text{prn: 53}\end{bmatrix}$$

$$\textit{pattern level}\quad \begin{bmatrix}\text{verb: }\beta\\ \text{cat: sp2}'\text{ a}'\text{ inf}\\ \text{fnc: order}\\ \text{arg: pro2 (}\alpha\text{ K)}\\ \text{prn: K+1}\end{bmatrix} \quad \begin{bmatrix}\text{noun: (}\alpha\text{ K)}\\ \text{fnc: }\beta\\ \text{prn: K+1}\end{bmatrix} \quad \begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix}$$

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

$$\textit{content level}\quad \begin{bmatrix}\text{verb: pass}\\ \text{cat: }\#\text{n-s3}'\ \#\text{a}'\text{ inf}\\ \text{fnc: order}\\ \text{arg: pro2 (salt 52)}\\ \text{prn: 53}\end{bmatrix} \begin{bmatrix}\text{noun: (salt 52)}\\ \text{cat: snp}\\ \text{fnc: pass}\\ \text{prn: 53}\end{bmatrix} \begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 53}\end{bmatrix}$$

Syntactically, the consequent is a standard infinitive construction with subject control.[9] There is also an interrogative-imperative adaptation, as from Did you pass the salt? to Tell me if you passed the salt!.

7.6 Repeated Hearsay

Instead of communicating a content from direct observation (6.3.2, 6.3.6) or from shadowing (6.6.4), a speaker may communicate a content which resulted from a hear mode interpretation. Let us call such a content *repeated*[10] *hearsay*. Repeated hearsay is a special constellation for production by the speaker and interpretation by the hearer.

For example, if Tweety tells Sylvester that Hector fell into the pool (STAR-1 content), this is stored by Sylvester as a STAR-2 content. It may in turn be told by Sylvester to Speedy as the new STAR-1 content Tweety said that Hector fell into the pool, which is in turn converted by Speedy into the new STAR-2 content Sylvester said that Tweety said that Hector fell into the pool.[11]

7.6.1 Time line of a repeated hearsay language content

tweety	*tweety*	*sylvester*	*sylvester*	*speedy*	
STAR–0	STAR–1	STAR–2	STAR–1	STAR–2	→ *time line*
origin	production	interpretation	production	interpretation	
e.g. direct observation	*conversion–1*	*conversion–2*	*conversion–1*	*conversion–2*	

Repeated STAR-1 STAR-2 instances may be iterated indefinitely, as in A said that B said that C said... (object clause iteration, 15.5.4; NLC Sect. 7.6). Consider the following example of a STAR-2 content interpreted by Sylvester and repeated as a STAR-1 content intended for Speedy:

7.6.2 STAR-1 content of Tweety said that Hector fell into pool.

sur: tweety	verb: say	verb: fall	sur: Hector	noun: pool	S: yard
noun: (bird x)	sem: past ind	sem: *that* past ind	noun: [dog x]	sem: *into*	T: 06/09/2013...
fnc: say	arg: (bird x) (fall 26)	fnc: (say 25)	fnc: fall	mdd: fall	A: sylvester
prn: 25	prn: 25	arg: [dog x]	prn: 26	prn: 26	R: speedy
		mdr: pool			3rd:
		prn: 26			prn: 25–26

[9] Infinitives are intrapropositional, in contradistinction to object clause constructions, which are extrapropositional.

[10] The legal terminology seems to be limited to *double* hearsay.

[11] Speedy may communicate Sylvester's STAR-1 content without taking the trouble of naming the sequence of sources by simply saying Hector fell into the pool. This loss of information regarding the origin makes the original content unassailable and turns it into a rumor (*on dit*).

The grammatical subject in a repeated hearsay content does not exclude first person, as shown by Sylvester said that Tweety said that I fell into the pool. There is also Sylvester said that I said that Tweety fell into the pool. Also, the verb of the final subclause is unrestricted with respect to tense, mood, and kind, as shown by Sylvester said that Tweety said that I could have fallen into the pool.

We conclude with the STAR-2 content as derived by the intended hearer Speedy, specified by the R value of the STAR-1 proplet in 7.6.2.

7.6.3 STAR-2 CONTENT OF Tweety said that Hector fell into pool.

sur: tweety	verb: say	verb: fall	sur: Hector	noun: pool	S: yard
noun: (bird x)	sem: past ind	sem: *that* past ind	noun: [dog x]	sem: *into*	T: 06/09/2013...
fnc: say	arg: (bird x) (fall 15)	fnc: (say 14)	fnc: fall	mdd: fall	A: speedy
prn: 14	prn: 14	arg: [dog x]	prn: 15	prn: 15	R: sylvester
		mdr: pool			3rd:
		prn: 15			prn: 14–15

The prn values of the STAR-2 content 7.6.3 differ from those of the STAR-1 content 7.6.2 because each agent has its own prn counter. This does not affect the grammatical nature of the extrapropositional relation between the two clauses, which happens to be a standard object clause construction.

8. STAR-2 Content: Hearer's Perspective

While a STAR-0 STAR-1 content conversion may apply to nonlanguage or language content, a hear mode STAR-1 STAR-2 conversion requires language. The STAR-1 content in a STAR-1 STAR-2 conversion is assumed insofar as it is the hearer's *interpretation* of the STAR-1 *surface*, which in turn depends on who is assumed to be the intended addressee (Sect. 10.1).

Sect. 8.1 presents the hearer's STAR-2 counterparts to the six speak mode STAR-1 variants which realize the past STAR-0 content 7.1.1. Sect. 8.2 describes the obligatory reinterpretation of indexicals in a STAR-1 STAR-2 conversion. Sects. 8.3–8.5 treat the hearer's revision of a speaker's syntactic mood adaptation, if present, into the literal original. Sect. 8.6 concludes with the adaptations needed for the imitation (self-performance) of an observed action.

8.1 From STAR-1 Surface to STAR-2 Perspective

The (i) origin, (ii) production, and (iii) interpretation of a content used in language communication are in an obligatory temporal order:

8.1.1 TIME LINE OF ORIGIN, PRODUCTION, AND INTERPRETATION

agent A		*agent A*		*agent B*	
	conversion−1		*conversion−2*		
STAR−0		STAR−1		STAR−2	
					time line
origin		production		interpretation	
I see you.		I saw you.		You saw me.	

In written language, the three stages may be arbitrarily far apart in time and space. The examples in this chapter, however, assume face-to-face communication in the medium of speech, which means that the stages are synchronized in parallel between the sensory modalities (11.2.1) of vocalization (speaker) and audition (hearer); without this synchronization of parallel production and interpretation, a modeling of natural dialog in speech is impossible.

Consider the hearer's STAR-2 counterparts to the STAR-1 variants in 7.1.2:

8.1.2 Pragmatically equivalent variants of STAR-1 contents

STAR-2 A: For nonlanguage content, there exists no hear mode counterpart.
STAR-2 B: Hector understands that Sylvester saw him.
STAR-2 C: Speedy understands that Sylvester saw "him," i.e. Hector.
STAR-2 D: Speedy understands that Sylvester saw Hector.
STAR-2 E: Speedy understands that Sylvester saw Hector on Thursday.
STAR-2 F: Speedy understands that Sylvester saw Hector on T. in the yard.

The hearer's STAR-1 STAR-2 conversions are realized explicitly as follows:

8.1.3 STAR-2 content B: You saw me derived from 7.1.4

			STAR-2 proplet of interpretation
sur:	sur:	sur:	S: veranda
noun: pro2	verb: see	noun: pro1	T: Sunday
cat: sp2	cat: #n-s3′ #a′ decl	cat: s1	A: hector
sem:	sem: past ind	sem: obq sg	R: sylvester
fnc: see	arg: pro2 pro1	fnc: see	3rd:
...	...	...	prn: 13
prn: 13	prn: 13	prn: 13	

The grammatical object pro1 points at the A-value hector and the grammatical subject pro2 points at the R-value sylvester.

8.1.4 STAR-2 content C: you saw him derived from 7.1.5

			STAR-2 proplet of interpretation
sur:	sur:	sur:	S: veranda
noun: pro2	verb: see	noun: pro3	T: Sunday
cat: sp2	cat: #n-s3′ #a′ decl	cat: snp	A: speedy
sem:	sem: past ind	sem: obq sg m	R: sylvester
fnc: see	arg: pro2 pro3	fnc: see	3rd: hector
...	...	...	prn: 13
prn: 13	prn: 13	prn: 13	

The subject pro2 points at the R-attribute sylvester, the grammatical object pro3 points at the 3rd-value hector, and the hearer is the A-value speedy,

8.1.5 STAR-2 content D: You saw Hector derived from 7.1.6

			STAR-2 proplet of interpretation
sur:	sur:	sur: hector	S: veranda
noun: pro2	verb: see	noun: [dog x]	T: Sunday
cat: sp2	cat: #n-s3′ #a′ decl	cat: snp	A: speedy
sem:	sem: past ind	sem: nm m	R: sylvester
fnc: see	arg: pro2 [dog x]	fnc: see	3rd:
dots	...	...	prn: 13
prn: 13	prn: 13	prn: 13	

The subject pro2 points at the R-value sylvester, the object hector is part of the content, which leaves the 3rd-slot empty, and the hearer is the A-value speedy.

While the S- and T-values in the variants A–D are lost in the STAR-0 STAR-1 transition of the speak mode, they may be preserved by the speaker by writing them as prepnouns into the content (variants E and F). In this way, they become available also in the STAR-1 STAR-2 transition of the hear mode:

8.1.6 STAR-2 CONTENT E: You saw Hector on T. (derived from 7.1.7)

				STAR-2 proplet
sur:	sur:	sur: hector	sur:	S: veranda
noun: pro2	verb: see	noun: [dog x]	noun: Thursday	T: Sunday
cat: sp2	cat: #n-s3′ #a′ decl	cat: snp	cat: snp	A: sylvester
sem:	sem: past ind	sem: nm m	sem: *on*	R: speedy
fnc: see	arg: pro2 [dog x]	fnc: see	fnc:	3rd:
mdr:	mdr: Thursday	mdr:	mdd: see	prn: 13
nc:	nc:	nc:	nc:	
pc:	pc:	pc:	pc:	
prn: 13	prn: 13	prn: 13	prn: 13	

This example resembles 8.1.5, except for preserving the time of origin (following in the content as a prepnoun).

8.1.7 STAR-2 CONTENT F: You saw Hector on T. in the y. (deriv. f. 7.1.8)

					STAR-2 proplet
sur:	sur:	sur: hector	sur:	sur:	S: veranda
noun: pro2	verb: see	noun: [dog x]	noun: Thursday	noun: yard	T: Sunday
cat: sp2	cat: #n-s3′ #a′ decl	cat: snp	cat: snp	cat: snp	A: sylvester
sem:	sem: past ind	sem: nm m	sem: *on* sg	sem: *in* def sg	R: speedy
fnc: see	arg: pro2 [dog x]	fnc: see	fnc:	fnc:	3rd:
mdr:	mdr: Thursday yard	mdr:	mdd: see	mdd: see	prn: 13
nc:	nc:	nc:	nc:	nc:	
pc:	pc:	pc:	pc:	pc:	
prn: 13	prn: 13	prn: 13	prn: 13	prn: 13	

This example resembles 8.1.6, except for preserving the location of origin (following in the content as another prepnoun).

8.2 Indexical STAR-1 STAR-2 Inference

The speaker's STAR-0 STAR-1 conversion-1 and the hearer's STAR-1 STAR-2 conversion-2 use different inferences for the interpretation of indexicals, namely 7.2.4 and 8.2.1. The reason is that the interpretation of pronominals in the speaker's conversion-1 is unchanged, but inverted in the hearer's conversion-2.

In the speak mode, the indexical inference 7.2.3 precedes a possible non-literal use; in the hear mode, the separate indexical inference 8.2.1 follows a reversion to literal use (mirror symmetry between speak and hear mode). The hear mode counterpart to the speaker’s indexical STAR-0 STAR-1 inference 7.2.3 is the following indexical STAR-1 STAR-2 inference:

8.2.1 INDEXICAL STAR-1 STAR-2 INFERENCE OF THE HEAR MODE

$$\begin{array}{lccccccc}
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: }\alpha\\ \text{arg: pro1 pro2}\\ \text{sem: }\zeta\text{ ind}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{S: L}'\\ \text{T: D}'\\ \text{A: P}\\ \text{R: N}\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix} & \Rightarrow \\
& \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow & \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: see}\\ \text{prn: 14}\end{bmatrix} &
\begin{bmatrix}\text{verb: see}\\ \text{sem: past ind}\\ \text{arg: pro1 pro2}\\ \text{prn: 14}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: see}\\ \text{prn: 14}\end{bmatrix} &
\begin{bmatrix}\text{S: yard}\\ \text{T: ...05T09:15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 14}\end{bmatrix} &
\begin{bmatrix}\text{S: study}\\ \text{T: 07/10/2013...}\\ \text{A: speedy}\\ \text{R: sylvester}\\ \text{3rd:}\\ \text{prn: 15}\end{bmatrix} &
\end{array}$$

$$\begin{array}{lcccc}
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: }\alpha'\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: }\alpha'\\ \text{arg: pro2 pro1}\\ \text{sem: }\zeta'\text{ ind}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: }\alpha'\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{S: L}'\\ \text{T: D}'\\ \text{A: P}\\ \text{R: N}\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix} \\
\multicolumn{5}{c}{\text{If D}' > \text{D, then }\zeta = \text{past. If D}' = \text{D,then }\zeta' = \zeta.} \\
& \Downarrow & \Downarrow & \Downarrow & \Downarrow \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: see}\\ \text{prn: 15}\end{bmatrix} &
\begin{bmatrix}\text{verb: see}\\ \text{sem: past ind}\\ \text{arg: pro2 pro1}\\ \text{prn: 15}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: see}\\ \text{prn: 15}\end{bmatrix} &
\begin{bmatrix}\text{S: study}\\ \text{T: 07/10/2013...}\\ \text{A: speedy}\\ \text{R: sylvester}\\ \text{3rd:}\\ \text{prn: 15}\end{bmatrix}
\end{array}$$

The inference takes speedy’s interpretation of sylvester’s STAR-1 surface I saw you, the assumed past STAR-1, and speedy’s current STAR-0 as input, and derives the STAR-2 content You saw me, with pro1 pointing at the R value sylvester and pro1 pointing at the A value speedy of the STAR-2 proplet. If the STAR-1 surface uses present tense, as in I see you, and the hearer’s interpretation is in close temporal proximity to the production (D = D′), the STAR-2 content uses present tense as well. The inferences 7.2.3 and 8.2.1 are alike in that they contain three STARs. They differ in that in 7.2.4 all three A-values are sylvester, but in 8.2.1 they are sylvester, speedy, speedy.

The systematic transitions (i) from an agent’s STAR-0 content to a speaker’s STAR-1 content and (ii) from a hearer’s interpretation of a STAR-1 surface to the hearer’s STAR-2 content regarding the literal perspective adjustments of indexicality may be illustrated as follows (reduced as compared to 7.1.9):

8.2.2 STAR-0 TO STAR-1 AND STAR-1 TO STAR-2 TRANSITIONS

	STAR-0	STAR-1	STAR-2
	one-place verb		
1	I walk	I walk	you walk
2		I walk	you walked
3		I walked	you walked
4	you walk	you walk	I walk
5		you walk	I walked
6		you walked	I walked
7	Peter walk	Peter walk	Peter walk
8		Peter walk	Peter walked
9		Peter walked	Peter walked
	two-place verb		
1	I see you	I see you	you see me
2		I see you	you saw me
3		I saw you	you saw me
4	you see me	you see me	I see you
5		you see me	I saw you
6		you saw me	I saw you
7	I see me	I see me	you see you
8		I see me	you saw you
9		I saw me	you saw you
10	I see Peter	I see Peter	you see Peter
11		I see Peter	you saw Peter
12		I saw Peter	you saw Peter
13	Peter see me	Peter see me	Peter see you
14		Peter see me	Peter saw you
15		Peter saw me	Peter saw you
16	Peter see Mary	Peter see Mary	Peter see Mary
17		Peter see Mary	Peter saw Mary
18		Peter saw Mary	Peter saw Mary
	three-place verb		
1	I give you flowers	I give you flowers	you give me flowers
2		I give you flowers	you gave me flowers
3		I gave you flowers	you gave me flowers
4	you give me flowers	you give me flowers	I give you flowers
5		you give me flowers	I gave you flowers
6		you gave me flowers	I gave you flowers
7	I give me flowers	I give me flowers	you give you flowers
8		I give me flowers	you gave you flowers
9		I gave me flowers	you gave you flowers
10	I give Suzy flowers	I give Suzy flowers	you give Suzy flowers
11		I give Suzy flowers	you gave Suzy flowers
12		I gave Suzy flowers	you gave Suzy flowers
13	Peter give me flowers	Peter give me flowers	Peter give you flowers
14		Peter give me flowers	Peter gave you flowers
15		Peter gave me flowers	Peter gave you flowers
16	Peter give Mary flowers	Peter give Mary flowers	Peter give Mary flowers
17		Peter give Mary flowers	Peter gave Mary flowers
18		Peter gave Mary flowers	Peter gave Mary flowers

In the first item of each triple, e.g. 1 2 3 or 4 5 6, the verb of the three contents is in present tense. In the second item, the verb of the first two contents is in the present tense, but past in the third, as when the speaker is live reporting and the hearer receives the sign later, for example when watching a rerun. In the third item, the verb of the first content is in the present tense (as always in monitoring), but the second content is in the past tense, as when the speaker is looking back, which makes the third content (hearer) past as well.

8.3 Reverting STAR-1 Interrogative into STAR-2 Imperative

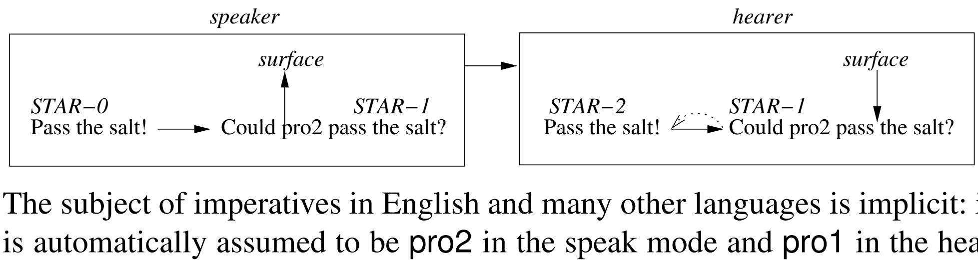
The mirror symmetry between production in the speak mode and interpretation in the hear mode holds for pronominal indexicals and is based on two different inferences in deductive use, namely 7.2.4 (speak mode) and 8.2.1 (hear mode). Another mirror symmetry holds for the nonliteral uses of (i) syntactic mood adaptation and (ii) figurative use, which each employ one and the same inference, but deductively in the speak and abductively in the hear mode.

Having shown the speak mode production of three syntactic mood adaptations in Sects. 7.3–7.5, each preceded by an adaptation of spatio-temporal indexicals, we turn in this and the next Sects. 8.3–8.5 to their hear mode counterparts, followed by the indexical STAR-1 STAR-2 inference 8.2.1. In the first example, produced in Sect. 7.3, the hearer is faced with the content of the STAR-1 surface Could you pass the salt? and must decide whether to interpret it as literal or as adapted.

If understood as literal, the hearer applies only the obligatory indexical inference 8.2.1, resulting in Could I pass the salt? In this case, the correct answer by the hearer (who might be in a body cast) would be yes or no. However, if Could you pass the salt? is understood as softening a command into a polite request, the hearer begins with reducing the content to the presumed original, followed by the indexical adjustment, resulting in (I) pass the salt! – which makes passing the salt the proper reaction.

Consider the speaker's adaptation combined with the hearer's reversal:

8.3.1 IMP-INT CONVERSION



The subject of imperatives in English and many other languages is implicit: it is automatically assumed to be pro2 in the speak mode and pro1 in the hear mode, without any surface manifestation. Consequently, the hearer's standard reversal from pro2 to pro1 is implicit as well.

In summary, a syntactic mood adaption in the speak mode begins with the (i) the obligatory STAR-0 STAR-1 inference 7.2.4 for spatio-temporal indexicality, (ii) followed by the deductive use of an optional inference for changing the syntactic mood (e.g. 13.1.3). The corresponding interpretation by the intended hearer begins with (iii) reverting the content of the STAR-1 surface into the original STAR-0 content by the data-driven abductive use of an inference such as 13.1.3, and (iv) followed by the obligatory adjustment of spatio-temporal and pronominal indexicality to the hear mode perspective by using the STAR-2 inference 8.2.1 deductively.

In successful communication, the input to the IMP-INT inference in the deductive use of the speak mode and the output of the abductive use of the hear mode are the same except for the prn numbering:

8.3.2 Hear mode using IMP-INT inference 13.1.3 abductively

$$
\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: sp2}'\text{ \#a}'\text{ vimp}\\ \text{arg: pro2 }\beta\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{noun: }\beta\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix}
\longrightarrow
\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: \#sp2}'\text{ \#a}'\text{ i.rog}\\ \text{sem: sbj }\textit{can}\\ \text{arg: pro2 (}\beta\text{ K)}\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{noun: }\beta\\ \text{cat: def sg}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix}
$$

If β ε {salt, pepper, potatoes}, then α = pass. If β ε {window, door}, then α ε {open, close}. ...

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

$$
\begin{bmatrix}\text{verb: pass}\\ \text{cat: sp2}'\text{ \#a}'\text{ vimp}\\ \text{sem: impv}\\ \text{arg: pro2 salt}\\ \text{prn: 25}\end{bmatrix}
\begin{bmatrix}\text{noun: salt}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 25}\end{bmatrix}
\begin{bmatrix}\text{S: yard}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 25}\end{bmatrix}
\begin{bmatrix}\text{verb: pass}\\ \text{cat: \#sp2}'\text{ \#a}'\text{ i.rog}\\ \text{sem: sbj }\textit{can}\\ \text{arg: pro2 (salt 24)}\\ \text{prn: 24}\end{bmatrix}
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: pass}\\ \text{prn: 24}\end{bmatrix}
\begin{bmatrix}\text{noun: salt}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 24}\end{bmatrix}
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 24}\end{bmatrix}
$$

The consequent takes the STAR-1 interrogative Could you pass the salt? as input, thereby binding α to *pass* and β to *salt*. With these values, which happen to be in the variable restrictions of α and β, the antecedent abductively derives the speaker's STAR-0 content (pro2) pass salt! in the hear mode.

It remains to apply the indexical STAR-1 STAR-2 inference 8.2.1 deductively to change the output from (pro2) pass salt! to (pro1) pass salt!:

8.3.3 Result of the hearer's STAR1 STAR-2 transition

$$
\begin{bmatrix}\text{verb: pass}\\ \text{cat: s1 \#a vimp}\\ \text{sem: impv}\\ \text{arg: pro1 salt}\\ \text{prn: 25}\end{bmatrix}
\begin{bmatrix}\text{noun: salt}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 25}\end{bmatrix}
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: speedy}\\ \text{R: sylvester}\\ \text{3rd:}\\ \text{prn: 25}\end{bmatrix}
$$

The result of the STAR-2 proplet adaptation in 8.3.1 inverts the A and R values of the STAR-1 proplet. Because the A value speedy is the pro1 value of the content, 8.3.3 is a blueprint for action. Whether or not Speedy actually realizes the blueprint is an additional step (subjunctive transfer, CLaTR Sect. 5.6) which may depend on the social relations between Speedy and Sylvester.

8.4 Reverting STAR-1 Declarat. into STAR-2 Interrogative

A hearer faced with the STAR-1 content I am asking you whether John passed the salt. (7.4.1) must decide whether it is literal or adapted. An interpretation as literal takes the content at face value and after the usual indexical adjustment, the content is left behind as information in the hearer's permanent memory, similar to Bill asked me whether John passed the salt.

However, if I am asking you whether John passed the salt. is interpreted as the optional strengthening adaptation of an interrogative into a declarative (Sect. 7.4), the correct reaction would be assent or dissent. The first step is applying the inference 7.4.2 abductively to the speaker's STAR-1 content:

8.4.1 Abductive hear mode use of INT-DECL inference 7.4.2

$$
\begin{array}{llllllll}
\textit{pattern level} &
\begin{bmatrix}\text{verb: }\alpha\\ \text{sem: interrog}\\ \text{arg: }\beta\ \gamma\\ \text{prn: K+2}\end{bmatrix} &
\begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\alpha\\ \text{prn: K+2}\end{bmatrix} &
\begin{bmatrix}\text{noun: (}\gamma\\ \text{fnc: pass}\\ \text{prn: K+2}\end{bmatrix} &
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K+2}\end{bmatrix} &
\leadsto &
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} \\
& \Downarrow & \Downarrow & \Downarrow & \Downarrow & & \Uparrow \\
\textit{content level} &
\begin{bmatrix}\text{verb: pass}\\ \text{sem: interrog past } \textit{do}\\ \text{arg: [person x] salt}\\ \text{prn: 26}\end{bmatrix} &
\begin{bmatrix}\text{sur: john}\\ \text{noun: [person x]}\\ \text{fnc: pass}\\ \text{prn: 26}\end{bmatrix} &
\begin{bmatrix}\text{noun: salt}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 26}\end{bmatrix} &
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 26}\end{bmatrix} &
&
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: ask}\\ \text{prn: 24}\end{bmatrix}
\end{array}
$$

$$
\begin{array}{llllll}
\begin{bmatrix}\text{verb: ask}\\ \text{sem: pres prog}\\ \text{arg: n}'\ \text{a}'\ (\alpha\ \text{K+1})\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: ask}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: }\alpha\\ \text{sem: } \textit{whether}\\ \text{arg: }\beta\ \gamma\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\alpha\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{noun: }\gamma\\ \text{fnc: }\alpha\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix} \\
\Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow \\
\begin{bmatrix}\text{verb: ask}\\ \text{cat: \#s1}'\ \text{\#sp2}'\ \text{decl}\\ \text{sem: pres prog}\\ \text{fnc: p1 p2 (pass 25)}\\ \text{prn: 24}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: ask}\\ \text{prn: 24}\end{bmatrix} &
\begin{bmatrix}\text{verb: pass}\\ \text{sem: } \textit{whether}\\ \text{arg: [p.x] (s. 23)}\\ \text{prn: 25}\end{bmatrix} &
\begin{bmatrix}\text{sur: john}\\ \text{noun: [p.x]}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 25}\end{bmatrix} &
\begin{bmatrix}\text{noun: salt}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 25}\end{bmatrix} &
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 24-25}\end{bmatrix}
\end{array}
$$

The object sentence construction in the consequent is mapped abductively back into Did John pass the salt? The STAR-2 of the consequent equals the STAR-2 of the antecedent, exept for the prn values. The STAR-2 values also equal the STAR-1 proplets of the speak mode counterpart 7.4.1 because of the joint dinner table situation and because the A and R values have not yet been inverted (8.4.3).

The result of the intermediate step may be shown as follows:

8.4.2 INTERMEDIATE CONTENT SERVING AS INPUT TO 8.2.1

$$\begin{bmatrix}\text{verb: pass}\\ \text{sem: interrog } \textit{do}\\ \text{arg: [person x] salt}\\ \text{prn: 26}\end{bmatrix}\begin{bmatrix}\text{sur: john}\\ \text{noun: [person x]}\\ \text{fnc: pass}\\ \text{prn: 26}\end{bmatrix}\begin{bmatrix}\text{noun: salt}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 26}\end{bmatrix}\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 26}\end{bmatrix}$$

The second step is the deductive application of the obligatory indexical STAR-1 STAR-2 inference 8.2.1. Because the content 8.4.2 does not contain any pronouns, the effect of the inference is limited to inverting the STAR's A and R values:

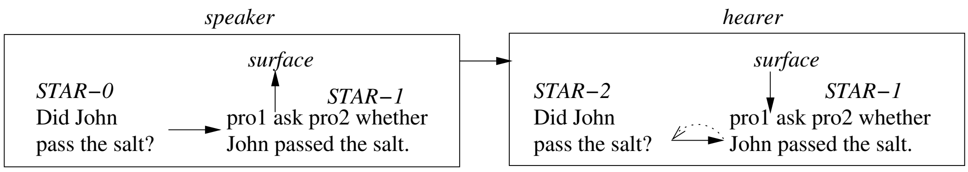
8.4.3 RESULT OF THE STAR-1 STAR-2 CONVERSION

$$\begin{bmatrix}\text{verb: pass}\\ \text{sem: interrog } \textit{do}\\ \text{arg: [person x] salt}\\ \text{prn: 26}\end{bmatrix}\begin{bmatrix}\text{sur: john}\\ \text{noun: [person x]}\\ \text{fnc: pass}\\ \text{prn: 26}\end{bmatrix}\begin{bmatrix}\text{noun: salt}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 26}\end{bmatrix}\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: speedy}\\ \text{R: sylvester}\\ \text{3rd:}\\ \text{prn: 26}\end{bmatrix}$$

At this point, Speedy understands the declarative I am asking you whether John passed the salt. as the literally used interrogative Did John pass the salt?, i.e. as a question which Sylvester would like to be answered (CLaTR Sect. 10.5).

Consider the speaker's adaptation combined with the hearer's reversal:

8.4.4 INT-DECL CONVERSION IN SPEAK AND HEAR MODE



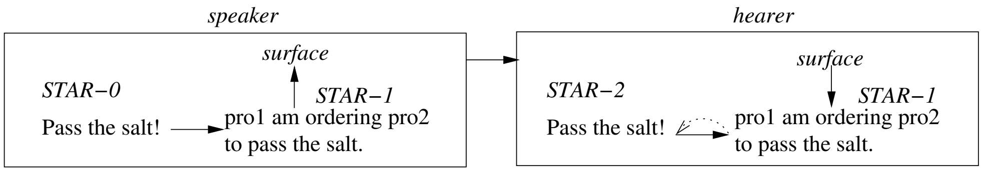
Syntactic mood adaptations in the speak and the hear mode rely on the data structure of proplets, which accommodates the full range of specificity from concrete constants to indexicals to restricted variables to values which are unrestricted except for what is required by their attribute.

8.5 Reverting STAR-1 Declarative into STAR-2 Imperative

A hearer faced with the STAR-1 content I am ordering you to pass the salt. must decide whether it is literal or adapted. If understood as literal, the content is taken at face value, similar to Mary ordered Jim to pass the salt., and stored in the hearer's memory as information without any call for action.

However, if I am ordering you to pass the salt. is interpreted as the result of the optional strengthening of an imperative into a declarative (7.5.1), the correct interpretation would be that the hearer should pass the salt:

8.5.1 IMP-DECL conversion in speak and hear mode



For the hearer to correctly understand the intent, the speaker's adaptation must be reverted to the original impulse. The first step is an abductive application of the inference 7.5.2, which derived the adaptive declarative deductively in the speak mode:

8.5.2 Using the IMP-DECL inference 7.5.2 abductively

$$\textit{pattern level}\quad \begin{bmatrix}\text{verb: }\alpha\\ \text{sem: impv}\\ \text{arg: pro2 }\beta\\ \text{prn: K+1}\end{bmatrix} \begin{bmatrix}\text{noun: }(\beta\text{ K})\\ \text{fnc: }\alpha\\ \text{prn: K+1}\end{bmatrix} \begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix} \rightsquigarrow \begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: order}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: order}\\ \text{sem: pres}\\ \text{arg: pro1 pro2 }\alpha\\ \text{prn: K}\end{bmatrix}$$

If β ε {salt, pepper, potatoes}, then α = pass. If β ε {window, door}, then α ε {open, close}, etc. $\Downarrow$ $\Downarrow$ $\Downarrow$ $\Uparrow$ $\Uparrow$

$$\textit{content level}\quad \begin{bmatrix}\text{verb: pass}\\ \text{cat: sp2}'\text{ \#a}'\text{ vimp}\\ \text{sem: impv}\\ \text{arg: pro2 (salt 23)}\\ \text{prn: 24}\end{bmatrix} \begin{bmatrix}\text{noun: (salt 23)}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 24}\end{bmatrix} \begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 24}\end{bmatrix} \quad \begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: order}\\ \text{prn: 23}\end{bmatrix} \begin{bmatrix}\text{verb: order}\\ \text{sem: pres}\\ \text{arg: pro1 pro2 pass}\\ \text{prn: 23}\end{bmatrix}$$

$$
\begin{array}{lllll}
\textit{pattern level} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: order}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: }\alpha\\ \text{fnc: order}\\ \text{arg: pro2 }\beta\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{noun: }\beta\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix}\\
& \Uparrow & \Uparrow & \Uparrow & \Uparrow\\
\textit{content level} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: order}\\ \text{prn: 23}\end{bmatrix} &
\begin{bmatrix}\text{verb: pass}\\ \text{fnc: order}\\ \text{arg: pro2 salt}\\ \text{prn: 23}\end{bmatrix} &
\begin{bmatrix}\text{noun: salt}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 23}\end{bmatrix} &
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 23}\end{bmatrix}
\end{array}
$$

As in 8.3.2 and 8.4.1, the prn values in this abductive use follow the overriding principle that the output of an extrapropositional operation must have a higher prn value than the input (Sect. 3.5; CLaTR 13.5.3).

The output of the abductive application corresponds to Pass the salt! and is defined as the following set of proplets (still prior to the second step in the hear mode, i.e. the indexical adjustment):

8.5.3 Intermediate content resulting from abductive 8.5.2

$$
\begin{bmatrix}\text{verb: pass}\\ \text{cat: sp2}'\text{ \#a}'\text{ vimp}\\ \text{sem: impv}\\ \text{arg: pro2 (salt 23)}\\ \text{prn: 24}\end{bmatrix}
\begin{bmatrix}\text{noun: (salt 23)}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 24}\end{bmatrix}
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 24}\end{bmatrix}
$$

The second step is the deductive application of the obligatory indexical STAR-1 STAR-2 inference 8.2.1. It changes implicit pro2 to implicit pro1 and inverts the STAR proplet's A and R values. The result is as follows:

8.5.4 Final result of the STAR-1 STAR-2 conversion

$$
\begin{bmatrix}\text{verb: pass}\\ \text{cat: s1 \#a vimp}\\ \text{sem: impv}\\ \text{arg: pro1 (salt 23)}\\ \text{prn: 24}\end{bmatrix}
\begin{bmatrix}\text{noun: (salt 23)}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: pass}\\ \text{prn: 24}\end{bmatrix}
\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: speedy}\\ \text{R: sylvester}\\ \text{3rd:}\\ \text{prn: 24}\end{bmatrix}
$$

At this point, Speedy understands the declarative I am ordering you to pass the salt. as the literally used imperative Pass the salt!, i.e. as the request that he should pass the salt to Sylvester (CLaTR Sect. 10.6).

8.6 Imitation: Using a Recognition Content for Action

The distinction between recognition and action applies not only to the interpretation of indexicals as they relate to a STAR-1 (speak mode, action) vs. a STAR-2 (hear mode, recognition), but also to imitation. It is defined as the transition from observation (recognition) to self-performance (action).

As an example from human[1] society, consider the agent Tom observing John brings Mary flowers and she[2] smiles as a STAR-0 content:

8.6.1 CONTENT OF John brings Mary flowers and she smiles

sur: John	sur:	sur: Mary	sur:
noun: [person x]	verb: bring	noun: [person y]	noun: flower
cat: snp	cat: #n′ #d′ #a′ decl	cat: snp	cat: pnp
sem: [person x]	sem: pres	sem: nm f sg	sem: indef pl
fnc: bring	arg: [person x] [person y] flower	fnc: bring	fnc: bring
mdr:	mdr:	mdr:	mdr:
nc:	nc: (smile 28)	nc:	nc:
pc:	pc:	pc:	pc:
prn: 27	prn: 27	prn: 27	prn: 27

sur: mary	sur:	S: park
noun: [person y]	verb: smile	T: Saturday
cat: snp	cat: #n′ decl	A: tom
sem: nm f sg	sem: pres	R:
fnc: smile	arg: [person y]	3rd: john, mary
mdr:	mdr:	prn: 27-28
nc:	nc:	
pc:	pc:	
prn: 28	prn: 28	

The two propositions are connected by the nc value (smile 28) of *bring*.

Motivated by the positive reaction of Mary, the observer Tom may want to self-perform (imitate) the action by applying it in an analogous constellation. The first step is to generalize the content into a pattern by replacing the grammatical subject and object with suitable indexicals and the prn values with variables, using simultaneous substitution:

[1] In natural agents without natural language, e.g. birds, imitation is probably the single most powerful learning mechanism between parents and their offspring.

[2] Representing coreferential (rather than indexical) she in the content 8.6.1 as a second *mary* proplet is reminiscent of early generative grammar, which treated coreference by means of coindexed copies, e.g. dog_i...dog_i, in the "deep structure" of sign-based PSG (Katz and Postal 1964). Instead of copying and co-indexing, DBS treats coreference by means of address.

8.6.2 Abstract pattern derived from content 8.6.1

noun: pro1	verb: bring	noun: pro3	noun: flower	noun: pro3	verb: smile
cat: snp	cat: #n′ #d′ #a′ decl	cat: snp	cat: pnp	cat: snp	cat: #n′ decl
sem: s1	sem: pres	sem: f sg obq	sem: indef pl	sem: s3	sem: pres
fnc: bring	arg: pro1 pro3 flw.	fnc: bring	fnc: bring	fnc: smile	arg: pro3
prn: K	nc: (smile K+1)	prn: K	prn: K	prn: K+1	prn: K+1
	prn: K				

The pattern is of medium concreteness (14.2.1) in that the prn values are variables and the grammatical subject and object are pronouns.

The pattern 8.6.2 may be turned into an inference (Sect. 14.3) by replacing the extrapropositional coordination connecting the two propositions with ⇒.

8.6.3 Extrapropositional coordination turned into inference

noun: pro1	verb: bring	noun: pro3	noun: flower		noun: pro3	verb: smile
cat: snp	cat: #n′ #d′ #a′ decl	cat: snp	cat: pnp		cat: snp	cat: #n′ decl
sem: s1	sem: pres	sem: f sg obq	sem: indef pl		sem: s3	sem: pres
fnc: bring	arg: pro1 pro3 flw.	fnc: bring	fnc: bring	⇒	fnc: smile	arg: pro3
prn: K	nc:	prn: K	prn: K		prn: K+1	prn: K+1
	prn: K					

This is called a C-inference (common sense inference, Sect. 5.1) because it has no set-theoretic aspect. It may be used to interpret an observed situation (recognition), but also for self-performance (action), as in pro1 bring luise flowers and luise smiles.

The action verb in the antecedent is *bring*. Action verbs are special in that they require two-sided rather than simple concept definitions, namely a side for recognizing an action and a side for self-performing it. The difference between the recognition and the action side is illustrated by easily recognizing another person's standing up on a floating surfboard, but having difficulty to self-perform this kind of action.[3]

In the self-performance of 8.6.3, the input to the antecedent of the inference is a content adjusted to the agent's own situation. Thereby the content to match the action verb pattern may not yet exist. In an application it may be reconstructed, however, from the available lexical proplet *bring* and its three available arguments (i) pro1 (required for self-performance), (ii) [person z] (provided by the agent's purpose), and (iii) flower (as the means to imitate the model provided by recognition):

[3] The other concepts in 8.6.4 have only simple definitions because the are equally suitable for use in recognition (observation) and action (imitation).

8.6.4 Attempt at self-performance of the inference 8.6.3

$$
\begin{array}{cccccccc}
\begin{bmatrix}\text{noun: }\alpha\\ \text{fnc: bring}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: bring}\\ \text{cat: \#n}'\text{ \#d}'\text{ \#a}'\text{ decl}\\ \text{arg: }\alpha\ \beta\text{ flower}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{noun: }\beta\\ \text{fnc: bring}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{noun: flower}\\ \text{cat: pnp}\\ \text{sem: indef pl}\\ \text{fnc: bring}\\ \text{prn: K}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{noun: }(\beta\text{ K})\\ \text{fnc: smile}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: smile}\\ \text{cat: \#n}'\text{ decl}\\ \text{arg: }(\beta\text{ K})\\ \text{prn: K+1}\end{bmatrix} \\
\Uparrow & \Downarrow & \Uparrow & \Uparrow & & \Downarrow & \Downarrow \\
\begin{bmatrix}\text{sur:}\\ \text{noun: pro1}\\ \text{cat: snp}\\ \text{sem: nm m}\\ \text{fnc: bring}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 27}\end{bmatrix} &
\mathbf{X} &
\begin{bmatrix}\text{sur: luise}\\ \text{noun: [p.z]}\\ \text{cat: snp}\\ \text{sem: nm f}\\ \text{fnc: bring}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 27}\end{bmatrix} &
\begin{bmatrix}\text{sur:}\\ \text{noun: flower}\\ \text{cat: pnp}\\ \text{sem: indef pl}\\ \text{fnc: bring}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 27}\end{bmatrix} &
&
\begin{bmatrix}\text{sur: luise}\\ \text{noun: [p.z]}\\ \text{cat: snp}\\ \text{sem: nm f}\\ \text{fnc: smile}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 28}\end{bmatrix} &
\begin{bmatrix}\text{sur:}\\ \text{verb: smile}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: fut}\\ \text{arg: [p.z]}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 28}\end{bmatrix}
\end{array}
$$

By connecting the three 'pillars of reference' pro1, [person z], and flower with the three-place verb bring in its action variant (⇓), the predicate of the self-performance may be instantiated as the following subjunctive:

8.6.5 Instantiating the **X** in 8.6.5

$$
\begin{bmatrix}\text{sur:}\\ \text{verb: bring}\\ \text{cat: \#n}'\text{ \#d}'\text{ \#a}'\text{ decl}\\ \text{sem: }\textit{can}\text{ pres subj}\\ \text{arg: pro1 [person z] flower.}\\ \text{mdr:}\\ \text{nc: (smile 28)}\\ \text{pc:}\\ \text{prn: 27}\end{bmatrix}
$$

Replacing X in 8.6.4 with 8.6.5 completes the input content for the antecedent of the inference. Binding β to [person z] enables the consequent to derive the expected result as the output. If the agent's final analysis evaluates[4] realization of the consequent as likely to enhance balance, the verbal mood of the antecedent is switched from subjunctive to imperative (CLaTR 5.6.1), resulting in a blueprint for action for possible realization by the interface component.

[4] At the neurological level, the mechanism of direct spontaneous evaluation has been studied in the fruit fly (Drosophila melanogaster) by Margulies et al. (2005). In higher organisms, evaluation is closely related to emotion (Tooby and Cosmides 1990, Ekman 1999, Lerner et al. 2015):

- Emotions function as pathways for fast, comprehensive access "to deal quickly with important interpersonal encounters." (Ekman 1999, p. 46)
- "Emotions are unbidden, not chosen by us." (Ekman op. cit. p. 54)
- The evaluation (appraisal) connected to an emotion may be influenced by the agent's phylo- and ontogenetical past (Lerner et al. 2015).

Despite the frequent co-occurrence of evaluation and emotion, they may be independent of each other. This is shown by such tasks as grading and reviewing, in which emotion need not, and should not, be involved. Also, evaluation may be reasoned or non-reasoned, as in rational vs. irrational likes and dislikes. In DBS, emotion is coded in the A slot of the STAR, e.g. [A: sylvester (happy)].

9. Pragmatics of Figurative Use

The nonliteral pragmatics of (i) syntactic mood adaptation[1] and (ii) figurative use have in common that they are based on inferences which apply deductively in the speak mode and abductively in the hear mode. They differ in that the input and output of syntactic mood inferences are complete propositions, while the input and output of figurative use inferences are parts of a proposition, elementary or phrasal.

Purposes of figurative use in the speak mode are (i) managing the contents resonating in the hearer and (ii) abbreviation. In nonlanguage cognition (3.1.3), figurative use allows the agent to view a content in a different light, which supports reasoning and influences behavior. The invariance constraint 6.4.4 ensures that the deductive output of a figurative use inference fits into the grammatical slot of the literal input,

9.1 Figurative Use of Referent I: Hyponymy

An example of using a noun concept figuratively is referring with The animal is tired to the dog sleeping by the stove. Animal is a hypernym (superordinate term) of {ape, bear, cat, dog, ...} and the concepts {ape, bear, cat, dog, ...} are hyponyms (subordinate terms) of animal. Hyponymy and hypernymy are classical lexical relations (Miller et al. 1993) which are symmetric in that for every hyponymy there is a corresponding hypernymy and vice versa.[2]

In DBS, hyponymy is formally treated as a lexical S-inference, i.e. an inference which has a set-theoretic aspect. The following definition implements hyponymy by associating several explicit sets of subordinate concepts with the names of the appropriate hypernyms:

---

[1] Sects. 7.3–7.5, 8.3–8.5.

[2] Other lexical pairs are synonymy (same meaning) and antonymy (opposite meaning), meronomy (parts of whole) and holonomy (whole of parts).

9.1.1 Lexical S-inference implementing hyponymy

$[\text{noun: } \alpha] \Rightarrow [\text{noun: } \beta]$

if $\alpha\ \varepsilon$ {ape, bear, cat, dog, ...}, then β is animal,
if $\alpha\ \varepsilon$ {apple, peach, pear, salad, ...}, then β is food,
if $\alpha\ \varepsilon$ {diesel, gasoline, electricity, hydrogen, ...} then β is fuel,
if $\alpha\ \varepsilon$ {bush, cactus, flower, grass, tree, vine, ...}, then β is plant,
...

The figurative use of animal for dog may be based either (i) on the hyponymy inference 9.1.1 alone, but deductively in the speak mode and abductively in the hear mode, or (ii) on the hyponymy inference 9.1.1 in the speak mode and the hypernymy inference 5.2.2 in the hear mode. Option (i) is more adequate empirically because it models the hearer's predicament of having to correctly select that particular element in the instantiation set (variable restriction on α in 9.1.1) from which the speaker proceeded to the hypernym, whereas two separate inferences merely offer two, albeit coextensive, sets of candidates.

Consider the speak mode application of the hyponymy inference:

9.1.2 Deductive use of hyponymy inference (speak mode)

inference:	α	$\Rightarrow$	β
	$\Uparrow$		$\Downarrow$
content:	dog be tired		animal be tired

The application of the inference 9.1.1 is data-driven. The initial trigger (13.5.1) is the input token dog. It is provided by recognition and binds the antecedent variable α ($\Uparrow$). The inference applies in the forward direction ($\Rightarrow$) to render the hypernym animal as the output ($\Downarrow$) of the consequent.

In the hear mode, the same inference is applied abductively to revert the figurative use back to the most likely primary referent:

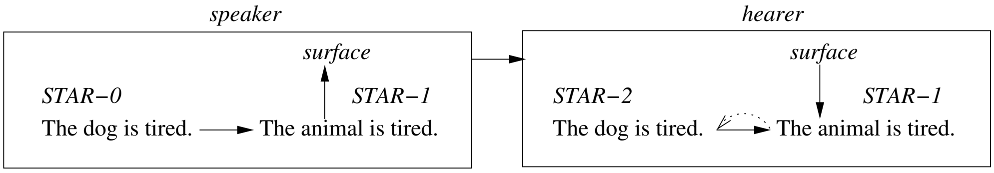
9.1.3 Abductive use of hyponymy inference (hear mode)

inference:	α		β
	$\Downarrow$		$\Uparrow$
content:	dog be tired		animal be tired

The initial trigger is the input token provided by automatic word form recognition, here animal, binding the variable β of the consequent ($\Uparrow$). The inference applies in the backward direction to render the possible instantiation dog as the output ($\Downarrow$) of the antecedent.

The interaction between the speaker's deductive and the hearer's abductive use of the hyponymy inference in language communication is integrated into the transition from a STAR-0 to a STAR-1 to a STAR-2 content and may be summarized graphically as follows:

9.1.4 FIGURATIVE USE BASED ON HYPONYMY RELATION



As a secondary coding, the speaker's figurative use adds subjective perspective to a content. For communication to succeed, however, the primary coding must be inferable for the hearer.

9.2 Figurative Use of Referent II: Property-Sharing

Another kind of figurative noun use has been called a *property sharing metaphor* (Glucksberg 2001 p. 58). Consider the following scenario: a hearer holding a coffee pot has just entered a room containing nothing but an orange crate. If the speaker requests Put the coffee on the table!, the hearer will infer that table is used to refer to the crate (FoCL Sect. 5.2).

However, if a prototypical table were standing next to the orange crate, the hearer would interpret the sentence differently, putting the coffee not on the crate but on the table. This is not caused by a change in the meaning$_1$ (6.1.1) of the word table, but by the fact that the context of use has changed, providing an additional and more suitable candidate for best match.

In contrast to the previous example (9.1.2, 9.1.3), there is no hypo-hyper relation between *orange crate* and *table*. Instead the nonliteral use of *table* is based on the following property-sharing inference:

9.2.1 PROPERTY-SHARING INFERENCE

α has flat surface $\Rightarrow$ α may serve as table

where α ε {dresser, side board, footstool, low bookshelf, orange crate, ...}

The speak mode uses 9.2.1 to go deductively from the antecedent to the consequent:

9.2.2 Deductive use of property-sharing inference

inference: α has flat surface $\Rightarrow$ α may serve as table
$\Uparrow$ $\Downarrow$
content: orange crate orange crate

The referent orange crate is bound to the variable α in the antecedent, enabling the consequent to characterize it as a possible table.

The hear mode, in contrast, uses 9.2.1 abductively to go from the consequent to the antecedent of the inference:

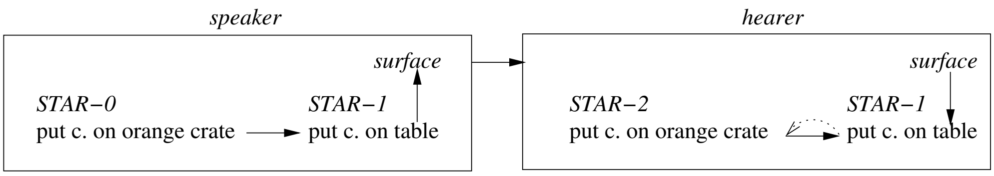
9.2.3 Abductive use of property-sharing inference

inference: α has flat surface $\dashrightarrow$ α may serve as table
$\Downarrow$ $\Uparrow$
content: orange crate orange crate

The concept *table* is part of the consequent. The abductive use provides the hearer with a reason of why the speaker might have selected the term table for an orange crate, which aids in finding the intended place (referent) for the coffee pot.

The speaker-hearer constellation may be shown graphically as follows:

9.2.4 Figurative use based on shared property inference



As in all abductive use, there is no certainty in the hear mode. For example, if the orange crate were accompanied by a dresser, a side board, a footstool, a commode, and a low bookshelf, it would be impossible for the hearer to decide what the speaker meant with table (too many candidates with flat horizontal surfaces in the current context of interpretation, embarrassment of riches). In such a case, the speaker would have to specify more precisely what is meant in order for communication to succeed.

9.3 Figurative Use of Referent III: Metonymy

A metonymy is based on a part-whole relation, as when the location Washington is used to refer to the US government, the feature pot belly to a man with an oversized girth, the abbreviation Mexico's renewable plants (CNN) to Mexico's renewable energy power plants, or the term Boston office to a person working in an office in Boston (Hobbs et al. 1993).

Like the figurative use of the hyponymy (Sect. 9.1) and the shared property relation (Sect. 9.2), a metonymy is based on a DBS inference which satisfies the invariance constraint 6.4.4 and is used deductively in the speak mode and abductively in the hear mode:

9.3.1 Part-whole inference

person working in $\beta \Rightarrow \beta$

The deductive use in the speak mode may be shown as follows:

9.3.2 Deductive use of metonymy inference

$$\begin{array}{llll} \textit{inference}: & \text{person working in } \beta & \Rightarrow & \beta \\ & \qquad\qquad\qquad\qquad\Uparrow & & \Downarrow \\ \textit{content}: & \text{person working in Boston office} & & \text{Boston office} \end{array}$$

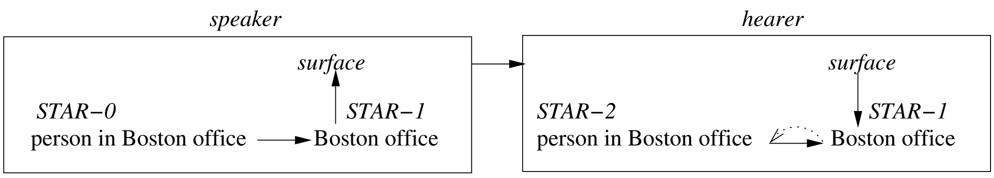
The hearer uses the same inference abductively:

9.3.3 Abductive use of metonymy inference

$$\begin{array}{llll} \textit{inference}: & \text{person working in } \beta & \rightarrow & \beta \\ & \qquad\qquad\qquad\qquad\Downarrow & & \Uparrow \\ \textit{content}: & \text{person working in Boston office} & & \text{Boston office} \end{array}$$

The hearer's abductive use restores the speaker's abbreviation derived in 9.3.2 into the long version. This is important for communication to succeed because otherwise the hearer could ask rhetorically Since when do office premises make phone calls?

9.3.4 Figurative use based on metonymy relation



The invariance constraint 6.4.4 is satisfied because Boston office and person working in the Boston office are syntactically both nouns and semantically both referents. Being in the same semantic field (11.3.3) is not required for a part-whole relation.

9.4 Figurative Use of Property I: Modifiers

Items of the Semantic kind (i) *referent* maintain identity over space and time, regardless of possible changes in size, shape, color, etc., as in the life cycle of a butterfly. The same identity is maintained in their nonliteral use, such as animal for dog, table for orange crate, or Boston office for the person working in the Boston office. Their use in communication is successful if, and only if, the hearer can determine the referent intended by the speaker.

The Semantic kind of (ii) *property*, in contrast, describes a quality of its modified which can freely change, as when an octopus changes from blue to brown. The corresponding Syntactic kinds are varied and include elementary adjectives in adnominal (beautiful) and adverbial (beautifully) use, prepnouns such as on the lake, and 1-place verbs such as melt. Without the need to establish identity, as in the Semantic kind referent, the figurative use of modifiers is free to serve other purposes.

One kind of using an adnominal modifier figuratively is abbreviation, as in the following example:

9.4.1 Abbreviating use of adnominal great

Booth shot a great man.

Referring to the assassination of President Lincoln by J.W. Booth in 1865, the use of great does not apply to the referent's physique (though Lincoln was tall), but to the phrasal modifier *with greater achievement than average*. The abbreviating use is based on the following inference:

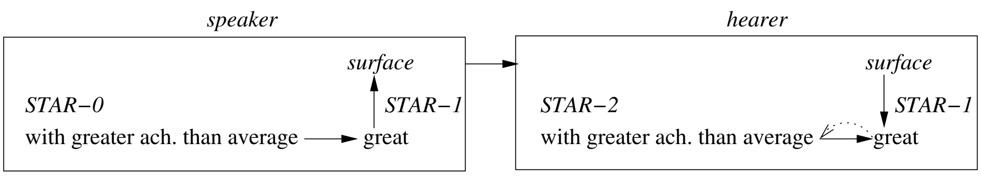
9.4.2 Inference coding abbreviating use of great

α with greater achievement than average $\Rightarrow$ great α

where α ε {artist, braggart, conman, composer, criminal, fraud, general, human being, imposter, leader, painter, poet, politician, man, musician, sculptor, woman, writer, ... }

Deductive use provides the STAR-0 STAR-1 conversion in the speak mode and abductive use the STAR-1 STAR-2 revision in the hear mode:

9.4.3 Speak and hear mode of an abbreviating adnominal use



The use of this inference obeys the invariance constraint 6.4.4, here between a literal phrasal and an elementary figurative modifier. Also, the speak mode applies the inference deductively, going forward from the antecedent to the consequent:

9.4.4 Deductive use of abbreviating adnominal inference

$$\begin{array}{lcr} \alpha \text{ with greater achievement than average} & \Rightarrow & \text{great } \alpha \\ \Uparrow & & \Downarrow \\ \textit{man} & & \textit{man} \end{array}$$

The hearer, in turn, binds *man* to α in the consequent of the inference to go abductively to the antecedent:

9.4.5 Abductive use of abbreviating adnominal inference

$$\begin{array}{lcr} \alpha \text{ with greater achievement than average} & \longrightarrow & \text{great } \alpha \\ \Downarrow & & \Uparrow \\ \textit{man} & & \textit{man} \end{array}$$

As in the other examples of figurative modifier use, the abbreviation applies to only part of the input proposition, here the phrase with greater achievement than average.

A modifier may be used adnominally, as in John drove a fast car (NLC Sect. 6.2) with fast modifying car, or adverbially, as in John drove fast with fast modifying drove (NLC Sect. 6.3). Abbreviating an adverbial modifier may be illustrated as follows:

9.4.6 Inference coding abbreviating use of have enough

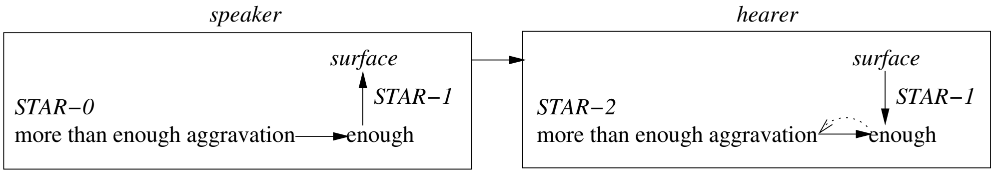
$$\alpha \text{ has more than enough of } \beta \Rightarrow \alpha \text{ has enough}$$

where α is an agent and β ε {aggravation, bigotry, conceit, corruption, cruelty, delay, depravity, dirt, dust, excuse, filth, fraud, greed, idiocy, impertinence, importunity, jealousy, lie, obtuseness, noise, poverty, quarrel, stupidity, treachery, violence, ...}

While having enough is normally a good thing, having more than enough may be paraphrased as having more than enough of a certain nuisance. The nature of the nuisance is well-manneredly left implicit: instead of naming the cause, it is left to the hearer to infer the nuisance in question from the context of use in quiet understanding.

As in the preceding examples, this abbreviating use is based on a deductive STAR-0 STAR-1 content conversion in the speak mode and an abductive STAR-1 STAR-2 content revision in the hear mode:

9.4.7 SPEAK AND HEAR MODE OF AN ABBREVIATING ADVERBIAL USE



The speaker uses the inference 9.4.6 deductively:

9.4.8 DEDUCTIVE USE OF ABBREVIATING ADVERBIAL INFERENCE

$$\begin{array}{llllll}
\alpha & \text{has more than enough of} & \beta & \Rightarrow & \alpha & \text{has enough} \\
\Uparrow & & \Uparrow & & \Downarrow & \\
\textit{fred} & & \textit{filth} & & \textit{fred} &
\end{array}$$

The value bound to the variable β in the antecedent is known to the speaker, but is omitted in the consequent.

The hearer uses the inference abductively:

9.4.9 ABDUCTIVE USE OF ABBREVIATING ADVERBIAL INFERENCE

$$\begin{array}{llllll}
\alpha & \text{has more than enough of} & \beta & \longleftrightarrow & \alpha & \text{has enough} \\
\Downarrow & & & & \Uparrow & \\
\textit{fred} & & & & \textit{fred} &
\end{array}$$

A complete interpretation of the antecedent requires the hearer to assign a value to the variable β. Based on the utterance situation, it is selected from the variable restriction in 9.4.6. In concord with Peirce, this abductive use is a guess at the best explanation (here the correct interpretation).

9.5 Figurative Use of Property II: 1-place verb

Compared to the Semantic kind referent, property and relation do not have *identity in change* (6.4.7). Without having to preserve the referent for the hearer in the transition from the literal to the nonliteral content, properties are free for the purpose of controlling the resonating content in the hearer, i.e. for a coloring of the interpretation without any clear literal counterpart.

As an example, consider the figurative use of the one-place verb (property) melt in the TV series NatGeo Wild *Secrets of Wild India* (2012):

9.5.1 METAPHORIC USE OF A PROPERTY

The tiger melts into the grass.

Clearly, the tiger is not melting literally. What is intended is a certain way of disappearing in the grass. The choice and the interpretation of melt may be related to the following semantic field:

9.5.2 SEMANTIC FIELD OF melt

{liquify, disappear, dissolve, fade away, melt, thaw, vanish, ...}.

As a fresh figurative use, 9.5.1 is not supported by convention (idiom) but by vivid TV images which dispel any uncertainty that might befall the viewer-hearer regarding the interpretation of melt. For communication to succeed, there is no clearly defined property for the hearer to identify because the speaker might have picked melt solely for coloring the scene with language.[3]

9.6 Figurative Use of Relation: 2- and 3-place Verbs

The uncertainty in the figurative uses of properties and relations due to the absence of a primary referent may be compensated by conventionalizing their interpretation as an idiom. An idiom supports the success of communication by favoring a learned nonliteral interpretation, yet complements the use with an aura of related concepts (NLC Sect. 5.6) provided by the semantic field (11.3.3.

Consider the following example:

9.6.1 FIGURATIVE USE OF THE 2-PLACE VERB steal

John stole the show.

Clearly, John doesn't steal literally, but one would be hard pressed to instantly provide a literal counterpart.[4]

[3] Whether or not nonliteral language (i) always has a literal counterpart which is (ii) required for understanding has been controversially discussed for more than two millennia (Ricoeur 1975). DBS takes a more differentiated approach by studying the Content kind concept with its computational Mechanism of type-token matching (Sects. 1.5, 6.4) and its instantiations as the Semantic kinds of referent (noun), property (adj, prepnoun, intransitive verb), and relation (transitive verbs).

[4] It is similar in the following examples of using *hit* (2-place verb, relation) nonliterally: The voters are hitting the polls, A cold beer hits the spot, Marilyn is hitting the booze again, Consumer prices hit 15 month high, and The marines hit the beach.

In contrast to 9.5.1, which is spontaneous but supported by a visual scene, 9.6.1 is conventionalized and supported by being an idiom. The origin of the idiom may be explained as the relation between (i) a content word used figuratively, here steal, and (ii) the associated *semantic field*:

9.6.2 SEMANTIC FIELD FOR FIGURATIVE USE OF steal show

α β show
where α is a person and β ε {kidnap, run off with, snap away, steal, take, usurp,...}

Without having to preserve a referent for the hearer in the transition from the literal to the nonliteral content, relations – like properties – are free for the purpose of controlling the resonating content in the hearer, i.e. for a coloring of the interpretation without any clear literal counterpart.

Another example of using a 2-place verb\object combination for a conventionalized nonliteral use is play the market:

9.6.3 FIGURATIVE USE OF play market

John plays the market.

Clearly, John doesn't play literally, but one would be hard pressed to instantly provide a literal counterpart. However, production and interpretation are supported by being conventionalized as an idiom. The origin of the idiom may be explained as the relation between (i) a content word used figuratively, here play, and (ii) the associated semantic field:

9.6.4 SEMANTIC FIELD FOR FIGURATIVE USE OF play market

α β market
where α is a person and β ε {amuse/entertain/enjoy/divert oneself with, gamble, have fun, play, ...}

Finally consider the figurative of a 3-place verb:

9.6.5 FIGURATIVE USE OF THE 3-PLACE VERB give

John gave Mary a headache.

The semantic field supporting the conventionalized nonliteral interpretation contains give as well as cause to have.

10. Higher-Level Pragmatics

There are phenomena of natural language communication which go beyond the standard DBS pragmatics of syntactic moods, sign kinds, and literal/nonliteral uses in the speak and hear mode as discussed so far. For example, when overhearing a dialogue between strangers, the indexical pro2 requires a STAR value which differs from a standard STAR-2 interpretation of an intended hearer (Sect. 10.1). Another case is the change of interpretation resulting from repeated reading of a paper in science (Sects. 10.2, 10.3). A wedding ceremony as a classic of ordinary language philosophy is reconstructed from the perspective of the official (speak mode) in Sect. 10.4 and of the couple getting married (hear mode) in Sect. 10.5. Sect. 10.6 concludes with a summary and classification of the 50 DBS inferences defined in the course of this book.

10.1 Overhearing

The pragmatic interpretation of a language surface depends crucially on whom it is intended for, as assumed by the hearer. For example, in overhearing it would not be appropriate to interpret the incoming surface as a STAR-1 content to be adapted to an intended hearer's STAR-2 perspective (8.2.1).

As an example, let us compare the interpretation of Are you hungry by the intended hearer Mary and the unintended hearer Tom. As members of the same language community, their different meaning$_2$ interpretations start out from the same meaning$_1$ (language content type, 6.2.3):

10.1.1 Are you hungry? AS A CONTENT TYPE (meaning$_1$)

$$\begin{bmatrix} \text{noun: pro2} \\ \text{fnc: be} \\ \text{prn: K} \end{bmatrix} \begin{bmatrix} \text{verb: be} \\ \text{cat: \#sp2}' \text{ \#be}' \text{ interrog} \\ \text{arg: pro2} \\ \text{mdr: hungry} \\ \text{sem: pres} \\ \text{prn: K} \end{bmatrix} \begin{bmatrix} \text{adj: hungry} \\ \text{mdd: be} \\ \text{prn: K} \end{bmatrix}$$

The intended hearer Mary begins with the meaning$_1$ interpretation of the speaker's surface as an attributed STAR-1 interpretation:

10.1.2 STAR-1 interpretation by intended hearer Mary

$$\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: be}\\ \text{prn: 16}\end{bmatrix}\begin{bmatrix}\text{verb: be}\\ \text{cat: \#ns1}'\ \text{\#be}'\ \text{interrog}\\ \text{arg: pro1}\\ \text{mdr: hungry}\\ \text{sem: pres}\\ \text{prn: 16}\end{bmatrix}\begin{bmatrix}\text{adj: hungry}\\ \text{mdd: be}\\ \text{prn:16}\end{bmatrix}\begin{bmatrix}\text{S: subway}\\ \text{T: Wednesday}\\ \text{A: john}\\ \text{R: mary}\\ \text{3rd:}\\ \text{prn: 16}\end{bmatrix}$$

Mary's STAR-1 proplet specifies the S, T, A, and R values as attributed to John, but assigns her own prn value.

Then Mary derives her STAR-2 interpretation by changing pro2 in the content to pro1, and inverting the A and R values:

10.1.3 STAR-2 interpretation by intended hearer Mary

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: be}\\ \text{prn: 16}\end{bmatrix}\begin{bmatrix}\text{verb: be}\\ \text{cat: \#ns1}'\ \text{\#be}'\ \text{interrog}\\ \text{arg: pro1}\\ \text{mdr: hungry}\\ \text{sem: pres}\\ \text{prn: 16}\end{bmatrix}\begin{bmatrix}\text{adj: hungry}\\ \text{mdd: be}\\ \text{prn:16}\end{bmatrix}\begin{bmatrix}\text{S: subway}\\ \text{T: Wednesday}\\ \text{A: mary}\\ \text{R: john}\\ \text{3rd:}\\ \text{prn: 16}\end{bmatrix}$$

Compared to the meaning$_1$ surface in the token 10.1.2, the content changes the indexical pro2 into pro1. Compared to Mary's STAR-1 interpretation 10.1.2, her STAR-2 proplet keeps the S and the T value, but changes the A value to the hearer mary, and the R value to the speaker john. In short, prompted by John, Mary is asking herself on Wednesday on the subway Am I hungry?

The interpretation by the unintended hearer Tom, in contrast, uses neither a STAR-1 nor STAR-2, but a STAR-0 proplet. Accordingly, he does not change the indexical in the content, but provides values in the 3rd slot of a STAR-0:

10.1.4 Unintended hearer interprets Are you hungry?

$$\begin{bmatrix}\text{noun: pro2}\\ \text{fnc: be}\\ \text{prn: 33}\end{bmatrix}\begin{bmatrix}\text{verb: be}\\ \text{cat: \#sp2}'\ \text{\#be}'\ \text{interrog}\\ \text{arg: pro2}\\ \text{mdr: hungry}\\ \text{sem: pres}\\ \text{prn: 33}\end{bmatrix}\begin{bmatrix}\text{adj: hungry}\\ \text{mdd: be}\\ \text{prn:33}\end{bmatrix}\begin{bmatrix}\text{S: subway}\\ \text{T: ...}\\ \text{A: tom (unintended hearer)}\\ \text{R:}\\ \text{3rd: stranger-1 stranger-2}\\ \text{prn: 33}\end{bmatrix}$$

As the value of the A attribute, tom is responsible for the content. Without an R value, the STAR is a STAR-0 instead of a STAR-2, whereby the indexical pro2 points at the value stranger-2 of the 3rd attribute.

The value hungry may coactivate (Sects. 3.2, 3.3) previous visits to various restaurants as resonating contents in Mary's as well as in Tom's memory. However, while those of Tom are irrelevant, those of Mary might not be, depending on whether or not her internal recognition (6.3.1) signals an appetite.

10.2 Pragmatics of Demanding Reading

Demanding reading, like a sophisticated literary work or a text from science, provides the pragmatic dimension of increasing understanding by an individual reader from repeated study. An example from physics is Einstein's (1920) essay on the theory of relativity. On page 9, Section III begins with the following account of classical mechanics (Newton 1667). Einstein puts it in double (scare[1]) quotes and proceeds to demolish it on almost 100 pages of print:

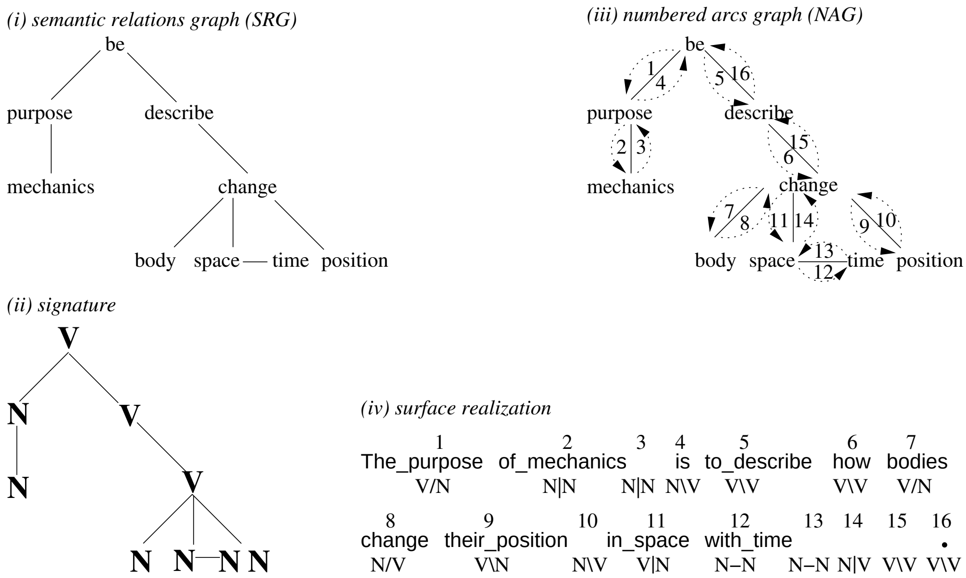
10.2.1 First sentence of Section III by Albert Einstein (1920)

> "THE PURPOSE of mechanics is to describe how bodies change their position in space with time."

Attentive reading of the complete text provides different contexts of interpretation, namely (i) a naive one before and (ii) better informed ones after study. A comparison of these interpretations shows the influence of resonating content in the hearer's memory on how the same content type (meaning$_1$, 6.1.1) may have different meaning$_2$ interpretations at different readings.

In DBS, a literal meaning$_1$ is built from (i) the core values and (ii) the semantic relations of structure between them. In the case of Einstein's sentence, the meaning$_1$ has the following canonical graph analysis (similar to 2.6.1):

10.2.2 Syntactic-semantic analysis of Einstein's sentence



---

[1] According to Carey McWilliams (1946 p. 298), scare quotes are quotation marks placed around a word or phrase to imply that it may not signify its apparent meaning or that it is not necessarily the

The semantic relations of structure are clearly coded into the language surface of the German original and carefully reproduced in the English translation.

More specifically, the noun *purpose* is the subject and the infinitive[2] *describe* is the object of the copula *be*. The prepnoun represented by *mechanics* modifies *purpose* adnominally.[3] The implicit subject of the infinitive *describe* is *purpose* (subject control[4]), the object is the adverbial clause[5] represented by the verb *change*. It takes *body* as subject and *position* as object. Furthermore, *change* is modified adverbially by the prepnoun coordination *in space with time*.

The semantic relations graph in 10.2.2 and the following set of proplets are related by a homomorphism (FoCL Sect. 21.3), i.e. for each elementary node in the SRG there is a corresponding proplet and for each relation in the SRG there is a corresponding relation between two proplets coded by address:

10.2.3 Proplet representation of Einstein's sentence

sur:	sur:	sur:	sur:
noun: purpose	noun: mechanics	verb: be	verb: describe
cat: snp	cat: adnv pnp	cat: #ns3′ #be′ decl	cat: #n′ #a′ inf
sem: def sg	sem: *of* indef pl	sem: *to* pres	sem: pres
fnc: be	mdd: purpose	arg: purpose describe	arg: (purpose) (change 77)
mdr: mechanics	mdr:	mdr:	fnc: be
nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:
prn: 76	prn: 76	prn: 76	prn: 76

sur:	sur:	sur:	sur:	sur:
verb: change	noun: body	noun: position	noun: space	noun: time
cat: #n-s3′ #a′ v	cat: pnp	cat: snp	cat: adnv snp	cat: adnv snp
sem: *how* pres	sem: indef pl	sem: poss sg	sem: *in* indef sg	sem: *with* indef sg
arg: body position	fnc: change	fnc: change	mdd: position	mdd: position
fnc: (describe 76)	mdr:	mdr: space	mdr:	mdr:
mdr: space time	nc:	nc:	nc: time	nc:
nc:	pc:	pc:	pc:	pc: space
pc:	prn: 77	prn: 77	prn: 77	prn: 77
prn: 77				

The relation between the verb *be* and the infinitive *describe* serving as its object is intrapropositional, as shown by their having the same prn value, here 76. The different prn values of *describe* and *change*, in contrast, show that the relation between the infinitive and its object sentence is extrapropositional. The prepositions *of, to, in, with* and the subordinating conjunction *how* are

way the quoting person would express its concept.

[2] TExer Sect. 4.4; CLaTR Sect. 8.4, 15.4; NLC 6.6.5–6.6.7.

[3] TExer Sect. 5.2; NLC 6.6.2–6.6.4, Sects. 15.1–15.4.

[4] TExer 4.4.4; CLaTR Sect. 15.4.

[5] TExer Sect. 3.5; NLC Sect. 7.5.

specified in the sem slots of *mechanics, describe, space, time*, and *change*, respectively.

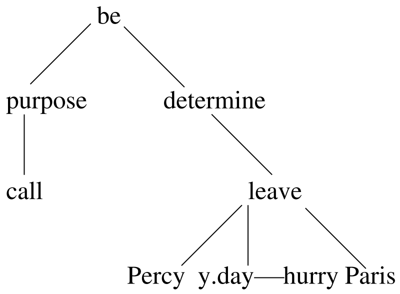
The contribution of the core values and the relations to the meaning$_1$ of a language expression may be shown by comparing 10.2.1 with an example of the same structure of syntactic/semantic relations, but different core values:

10.2.4 VARIATION OF SENTENCE 10.2.1 WITH DIFFERENT CORE VALUES

The purpose of the call was to determine why Percy left Paris yesterday in a hurry.

The grammatical similarity between the sentences 10.2.1 and 10.2.4 is shown by comparing the SRG of 10.2.2 with the following variant:

10.2.5 VARIANT OF THE SEMANTIC RELATIONS GRAPH 10.2.2



Like Einstein's sentence 10.2.1, the grammatical construction consists of the copula *be* connecting a noun as subject (*purpose*) and an infinitive as object (*describe* → *determine*). The subject takes a prepnoun (*of mechanics* → *of the call*) as an adnominal modifier. The infinitive takes an object sentence (*change* → *leave*) as its argument. The transitive verb of the object sentence takes a noun as subject (*body* → *Percy*) and a noun as object (*position* → *Paris*). The verb is modified by time (*yesterday*) conjoined with manner (*hurry*). The homomorphic proplet representation of 10.2.5 (not shown) corresponds to 10.2.3 modulo different core, continuation, cat, sem, and prn values.

10.3 How Repeated Study Changes Interpretation

Thanks to language acquisition, the core values and the kinds of semantic relations of a well-formed grammatical structure are shared more or less[6] by

[6] Harbsmeier (2005) describes the complications lurking behind the notion of a *natural language* from a diachronic and a synchronic point of view, using examples from Chinese, Russian, Italian, Arabic, and other languages.

all members of a natural language community. Thus the literal meanings$_1$ of the sentences 10.2.1 and 10.2.4 as types are each the same for the majority of agents competent in English. The meaning$_2$, i.e. individual understanding of the speaker, in contrast, will vary. This is because individual contexts of interpretation (Sect. 6.1) differ in experience, knowledge, and belief.

Instead of arguing directly[7] against the meaning$_1$ (content type) defining the purpose of classical mechanics in 10.2.1, Einstein aims at rebuilding the resonating content in the reader's memory by explicating what should be meant by space and time *exactly*. This is formulated in the sentence directly following 10.2.1, now without double quotes:

10.3.1 Continuing after sentence 10.2.1

> I should load my conscience with grave sins against the sacred spirit of lucidity were I to formulate the aims of mechanics in this way, without serious reflection and detailed explanations.

On each reading, these reflections and explanations have the effect of extending the reader's individual understanding, coactivated automatically by the owner values space and time (Sect. 3.2; CLaTR Sect. 5.4).

Thus, to explain how to understand Einstein's sentence 10.2.1 as intended we must distinguish systematically between the literal meaning$_1$ of the language sign and the speaker meaning$_2$ of the utterance (6.1.1). In an individual reader, successive readings of Einstein's sentence on page 9 do not change the sparse literal meaning$_1$, but the meaning$_2$ may change because of new content which resonates with each re-reading of 10.2.1, thereby extending and correcting the original interpretation.

The crucial new content changing the interpretation of 10.2.1 follows 23 pages later in the text:

10.3.2 Sentence on p. 32 of Einstein (1920)

> Every reference-body has its own particular time.

This content shades the interpretation of 10.2.1 by denying Newton's assumption that bodies changing their position in space all run on the same time.

To formally show how the content 10.3.2 resonates with the content 10.2.3, consider its representation as a set of proplets:

---

[7] As when pointing out the error in 2+2=5.

10.3.3 PROPLET REPRESENTATION OF 10.3.2

sur:	sur:	sur:	sur:	sur:	sur:
noun: body	noun: reference	verb: have	noun: time	adj: own	adj: particular
cat: snp	cat: sn	cat: #n-s3′ #a′ decl	cat: snp	cat:	cat:
sem: pl exh	sem:	sem: pres	sem:	sem:	sem:
fnc: have	fnc:	arg: body time	fnc: have	mdr: particular	mdr:
mdr: reference	mdd: body	mdr:	mdr: own	mdd: time	mdd: own
nc:	nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:	pc:
prn: 95	prn: 95	prn: 95	prn: 95	prn: 95	prn: 95

Because A-memory stores content in the order of arrival, the (i) content of the first reading (10.2.1, prn: 76), the (ii) additional content (10.3.3, prn: 95), and (iii) reading 10.2.1 a second time (prn: 103) are in the following order:

10.3.4 ADDITIONAL CONTENT BETWEEN FIRST AND SECOND READING

member proplets		*now front*
first reading of 10.2.1	additional content 10.3.2	second reading of 10.2.1
[prn: 76]	[prn: 95]	[prn: 103]

While the first reading of 10.1.2 (prn value 76) can not resonate with the additional content 10.3.2 (prn value 95) because it precedes, the second reading (prn value 103) does because it follows. By remembering the first and the second pragmatic meaning$_2$ interpretation, the agent may compare different degrees of understanding, which is an important aspect of conscious thought.

10.4 Language Ritual: Speak Mode

Speech Act Theory (Austin [1955]1962) analyzes the use of certain formulas in language rituals such as the baptism of a child or a ship, sentencing in court, or performance of a wedding ceremony. Their point is not so much understanding the formula correctly, but the validity of the ritual in society.

In a wedding ceremony, for example, the speaker must be an authorized official and the literal STAR-0 content must be expressed as a STAR-1 formula:[8]

10.4.1 SPEAK MODE: STAR-0 STAR-1 CONVERSION

STAR-0 input: pro1 marry pro2
STAR-1 output: pro1 now pronounce pro2 man and wife

[8] In addition to the proper performance of the ceremony, validity usually depends on additional conditions which apply to dowry, religion, gender, age, etc., and vary between countries and cultures.

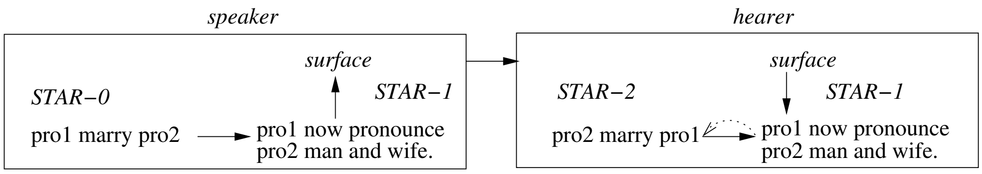
In the hear mode, the couple getting married reverts the content of the STAR-1 surface back into the speaker's presumed STAR-0 content, but with the adaptation of indexicals to the hear mode perspective, resulting in a STAR-2 content:

10.4.2 HEAR MODE: STAR-1 STAR-2 CONVERSION

STAR-1 input: pro1 now pronounce pro2 man and wife
STAR-2 output: pro2 marry pro1 (You marry us)

The combination of the two conversions may be shown graphically as follows:

10.4.3 STAR-0 STAR-1 AND STAR-1 STAR-2 CONVERSIONS



In the speak mode, the content matching the antecedent is a nonlanguage token (6.2.2), the subject is pro1, the verb is marry in the present indicative, and the object is pro2. In the language content derived by the consequent of the wedding inference 10.4.4, the subject and the first object are unchanged, but the verb marry is replaced by pronounce which takes the noun conjunction man and wife as a second object.[9]

In DBS, the wedding ceremony is implemented as the following inference, shown as it applies deductively in the speak mode:

10.4.4 SPEAKER'S DEDUCTIVE APPLICATION OF WEDDING INFERENCE

$$
\begin{array}{llllllll}
\textit{pattern level} &
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: marry}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{verb: marry}\\ \text{cat: \#n-s3}'\text{ \#a}'\text{ decl}\\ \text{sem: pres}\\ \text{arg: pro1 pro2}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: marry}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: Mons.}\\ \text{R: }\alpha\&\beta\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{sur: I}\\ \text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: pronounce}\\ \text{prn: K+1}\end{bmatrix} \\
& \Uparrow & \Uparrow & \Uparrow & \Uparrow & & \Downarrow \\
\textit{content level} &
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: marry}\\ \text{prn: 11}\end{bmatrix} &
\begin{bmatrix}\text{verb: marry}\\ \text{cat: \#n-s3}'\text{ \#a}'\text{ decl}\\ \text{sem: pres}\\ \text{arg: pro1 pro2}\\ \text{prn: 11}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: marry}\\ \text{prn: 11}\end{bmatrix} &
\begin{bmatrix}\text{S: church}\\ \text{T: Saturday}\\ \text{A: Mons.}\\ \text{R: bob\&bess}\\ \text{3rd:}\\ \text{prn: 11}\end{bmatrix} &
&
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: pronounce}\\ \text{prn: 12}\end{bmatrix}
\end{array}
$$

[9] The verb pronounce shows that the objects of three place verbs in English need not always be an indirect and a direct object, but may both be direct objects (accusatives), e.g. you (acc) and man and wife (acc) in 10.4.1. This is similar to the German verb lehren (teach) taking two accusative objects, as in Maria lehrte den Freund (acc) den Foxtrott (acc) ∼ Mary taught the friend the foxtrot.

$$
\begin{array}{llllll}
\textit{pattern level} &
\begin{bmatrix}\text{sur: pronounce}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ \#a\&}' \text{ decl}\\ \text{sem: } \textit{now} \text{ pres}\\ \text{verb: pronounce}\\ \text{arg: pro1 pro2 man\&}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{sur: you}\\ \text{cat: sp2}\\ \text{noun: pro2}\\ \text{fnc: pronounce}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{sur: man}\\ \text{noun: man\&}\\ \text{fnc: pronounce}\\ \text{sem: } \textit{and}\\ \text{nc: wife}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{sur: wife}\\ \text{noun: wife}\\ \text{fnc:}\\ \text{pc: man\&}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: Mons.}\\ \text{R: } \alpha\&\beta\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix}\\
& \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow\\
\textit{content level} &
\begin{bmatrix}\text{verb: pronounce}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ \#a\&}' \text{ decl}\\ \text{sem: } \textit{now} \text{ pres}\\ \text{arg: pro1 pro2 man\&}\\ \text{prn: 12}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: pronounce}\\ \text{prn: 12}\end{bmatrix} &
\begin{bmatrix}\text{noun: man\&}\\ \text{fnc: pronounce}\\ \text{sem: } \textit{and}\\ \text{nc: wife}\\ \text{prn: 12}\end{bmatrix} &
\begin{bmatrix}\text{noun: wife}\\ \text{fnc:}\\ \text{pc: man\&}\\ \text{prn: 12}\end{bmatrix} &
\begin{bmatrix}\text{S: church}\\ \text{T: Saturday}\\ \text{A: Mons.}\\ \text{R: bob\&bess}\\ \text{3rd:}\\ \text{prn: 12}\end{bmatrix}
\end{array}
$$

The STAR-0 and STAR-1 contents with their indexicals adapted to the speak mode perspective equal 10.4.1. The indexical pro1 points at the A value Mons.(ignore) and pro2 at the R values bob&bess of the STARs. The adverbial is coded as the value *now* in the sem slot of *pronounce*.

10.5 Language Ritual: Hear Mode

Similar to the hear mode conversions in Sects. 8.3–8.5, the engaged couple's STAR-1 STAR-2 hear mode inferencing requires two steps. The first is the abductive application of the wedding inference 10.4.4. The second is the deductive application of the indexical STAR-1 STAR-2 inference 8.2.1; it replaces pro1 with pro2 and pro2 with pro1, and inverts the A and R values in the STAR-2 (mirror symmetry between speak and hear mode, Sect. 8.2).

The abductive application of 10.4.4 may be shown as follows:

10.5.1 HEARER'S ABDUCTIVE APPLICATION OF WEDDING INFERENCE

$$
\begin{array}{llllllll}
\textit{pattern level} &
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: marry}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{verb: marry}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ decl}\\ \text{sem: pres}\\ \text{arg: pro1 pro2}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: marry}\\ \text{prn: K+1}\end{bmatrix} &
\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: Mons.}\\ \text{R: } \alpha\&\beta\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix} &
\longrightarrow &
\begin{bmatrix}\text{sur: I}\\ \text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: pronounce}\\ \text{prn: K}\end{bmatrix}\\
& \Downarrow & \Downarrow & \Downarrow & \Downarrow & & \Uparrow\\
\textit{content level} &
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: marry}\\ \text{prn: 6}\end{bmatrix} &
\begin{bmatrix}\text{verb: marry}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ decl}\\ \text{sem: pres}\\ \text{arg: pro1 pro2}\\ \text{prn: 6}\end{bmatrix} &
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: marry}\\ \text{prn: 6}\end{bmatrix} &
\begin{bmatrix}\text{S: church}\\ \text{T: Saturday}\\ \text{A: Mons.}\\ \text{R: bob\&bess}\\ \text{3rd:}\\ \text{prn: 6}\end{bmatrix} &
&
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: pronounce}\\ \text{prn: 5}\end{bmatrix}
\end{array}
$$

$$
\begin{array}{lccccc}
\textit{pattern} & \begin{bmatrix}\text{sur: pronounce}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ \#a\&}' \text{ decl}\\ \text{sem: } \textit{now} \text{ pres}\\ \text{verb: pronounce}\\ \text{arg: pro1 pro2 man\&}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{sur: you}\\ \text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: pronounce}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{sur: man}\\ \text{noun: man\&}\\ \text{fnc: pronounce}\\ \text{sem: } \textit{and}\\ \text{nc: wife}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{sur: wife}\\ \text{noun: wife}\\ \text{fnc:}\\ \text{pc: man\&}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: Mons.}\\ \text{R: } \alpha\&\beta\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix}\\
\textit{level} & & & & & \\
 & \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow\\
\textit{content} & \begin{bmatrix}\text{verb: pronounce}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ \#a\&}' \text{ decl}\\ \text{sem: } \textit{now} \text{ pres}\\ \text{arg: pro1 pro2 man\&}\\ \text{prn: 5}\end{bmatrix} & \begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: pronounce}\\ \text{prn: 5}\end{bmatrix} & \begin{bmatrix}\text{noun: man\&}\\ \text{fnc: pronounce}\\ \text{sem: } \textit{and}\\ \text{nc: wife}\\ \text{prn: 5}\end{bmatrix} & \begin{bmatrix}\text{noun: wife}\\ \text{fnc:}\\ \text{pc: man\&}\\ \text{prn: 5}\end{bmatrix} & \begin{bmatrix}\text{S: church}\\ \text{T: Saturday}\\ \text{A: Mons.}\\ \text{R: bob\&bess}\\ \text{3rd:}\\ \text{prn: 5}\end{bmatrix}\\
\textit{level} & & & & &
\end{array}
$$

This abductive application differs from the deductive use in 10.4.4 because of the inversion of the $\Uparrow$ and the $\Downarrow$, and use of the $\dashrightarrow$ connective. The prn values differ from the deductive use because every agent has its own prn counter.

Finally, the indexical STAR-1 STAR-2 inference 8.2.1 applies deductively with the following result (only content level shown):

10.5.2 Deductive STAR-2 indexical adjustment

$$
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: s1}\\ \text{fnc: marry}\\ \text{prn: 6}\end{bmatrix}
\begin{bmatrix}\text{verb: marry}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ decl}\\ \text{sem: pres}\\ \text{arg: pro1 pro2}\\ \text{prn: 6}\end{bmatrix}
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: oblique}\\ \text{sem: pl}\\ \text{fnc: marry}\\ \text{prn: 6}\end{bmatrix}
\begin{bmatrix}\text{S: church}\\ \text{T: Saturday}\\ \text{A: Mons.}\\ \text{R: bob\&bess}\\ \text{3rd:}\\ \text{prn: 6}\end{bmatrix}
\Rightarrow
$$

$$
\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: marry}\\ \text{prn: 6}\end{bmatrix}
\begin{bmatrix}\text{verb: marry}\\ \text{cat: \#n-s3}' \text{ \#a}' \text{ decl}\\ \text{sem: pres}\\ \text{arg: pro2 pro1}\\ \text{prn: 6}\end{bmatrix}
\begin{bmatrix}\text{noun: pro1}\\ \text{cat: oblique}\\ \text{sem: pl}\\ \text{fnc: marry}\\ \text{prn: 6}\end{bmatrix}
\begin{bmatrix}\text{S: church}\\ \text{T: Saturday}\\ \text{A: bob\&bess}\\ \text{R: Mons.}\\ \text{3rd:}\\ \text{prn: 6}\end{bmatrix}
$$

In the deductive output, the official performing the ceremony is pro2, the couple getting married is pro1, and the A and R values of the STAR-1 input proplet are inverted in the STAR-2 output proplet.

The STAR-2 content specifies the purpose of the STAR-1 content, i.e. what the authorized official intended with it. Within the conventions of the culture, this may seem self-evident, but an easy, well-practiced STAR-1 STAR-2 content conversion does not make it less of a reasoning. For example, for someone who has not yet learned the wedding inference, e.g. a child attending the ceremony for the first time, the STAR-2 output content would not be obvious and an explanation would be informative.

10.6 List of DBS Inferences

Let us conclude Part II by sorting the DBS inference schemata, inferences, and inference applications presented in the course of this text into 10 different classes. The classes roughly follow the order of their appearance.

The first class contains the schemata of the logical S-inferences. They are so-called because they have a set-theoretical aspect which is used in the DBS reconstruction of some classical deduction rules (syllogisms).

10.6.1 CONSTELLATIONS OF LOGICAL S-INFERENCES

1-3 Unnegated antecedent and unnegated consequent (4.1.6)
4-6 Unnegated antecedent and negated consequent (4.1.7)
7-9 Negated antecedent and unnegated consequent (4.1.8)
10-12 Negated antecedent and negated consequent (4.1.9)

In accordance with classical tradition, each of these constellations is analyzed in a universal, a particular, and an individual variant. The reconstructions of 8 classical deduction rules within these schemata are listed in 10.6.4.

The second class contains lexical S-inferences. They are so-called because they are based on set-theoretic structures in the lexicon.

10.6.2 LEXICAL S-INFERENCES

13. S-inference implementing a synonymy (5.2.3)
14. S-inference implementing an antonymy (5.2.4)
15. S-inference implementing a hyponymy (9.1.1)
16. S-inference implementing a hypernymy (5.2.2)

The third class are C-inferences which are so-called because they are based on common sense reasoning, without relying on set-theory.

10.6.3 C-INFERENCES

17. C-inference illustrating deductive and abductive use (3.5.1, 3.5.2)
18. C-inference implementing a cause and effect relation (5.1.2)
19. C-Inference implementing a counter measure (5.1.4)
20. C-inference creating a summary (5.5.2)

The fourth class are the classical syllogisms analyzed as DBS inferences:

10.6.4 Inferences of logical deduction

21. modus ponendo ponens (4.2.3)
22. modus tollendo tollens (4.3.4)
23. modus BARBARA (4.4.4)
24. modus CELARENT (4.4.9)
25. modus DARII (4.5.4)
26. modus FERIO (4.5.9)
27. modus BAROCO (4.6.4)
28. modus BOCARDO (4.6.9)

The fifth class contains the two inferences for adjusting indexicals to the different perspectives of the speak and the hear mode. Their application is obligatory.

10.6.5 Inferences for indexical adaptation

29. indexical STAR-0 STAR-1 inference in the speak mode (7.2.3)
30. indexical STAR-1 STAR-2 inference in the hear mode (8.2.1)

The sixth class contains inferences which (i) change the syntactic mood of a content to adapt it to the speaker's communicative purpose (deductive use) and (ii) reverse the adaptation in the hear mode (abductive use) to reveal the original intent.

10.6.6 Inferences for syntactic mood adaptation

31. Deductive use of IMP-INT inference for STAR-0 STAR-1 adaption (13.1.3)
32. Deductive use of INT-DECL inf. for STAR-0 STAR-1 adaption (7.4.2)
33. Deductive use of IMP-DECL inf. for STAR-0 STAR-1 adaption (7.5.2)
34. Abductive use of IMP-INT inference for STAR-1 STAR-2 adaption (8.3.2)
35. Abductive use of INT-DECL inf. for STAR-1 STAR-2 adaption (8.4.2)
36. Abductive use of IMP-DECL inf. for STAR-1 STAR-2 adaption (8.5.2)

These inferences are optional in the speak mode, but obligatory in the hear mode if they were applied in the speak mode (i.e. if the speaker did not mean the utterance literally).

The seventh class contains inferences which (i) change an original content by inserting figurative uses to adapt it to the speaker's communicative purpose and (ii) reverse the adaptation in the hear mode to reveal the original content.

10.6.7 Inferences for Figurative use

37. Deductive use of hyponymy inference (9.1.2)
38. Abductive use of hyponymy inference (9.1.3)
39. Deductive use of property sharing inference (9.2.2)
40. Abductive use of property sharing inference (9.2.3)
41. Deductive use of part-whole inference (9.3.2)
42. Abductive use of part-whole inference (9.3.3)
43. Inference for figurative use of steal (9.4.2)
44. Inference for figurative use of play (9.4.3)
45. Inference coding abbreviating use of great (9.6.2)
46. Inference coding abbreviating use of have enough (9.4.6)

The eighth class contains the inference for performing a wedding ceremony.

10.6.8 Explicit performative inference

47. Official's deductive speak mode application of wedding inference (10.4.4)
48. Engaged couple's abductive hear mode application of wedding inference (10.5.1)

Numerous other constellations for performing social functions, like a judge pronouncing a sentence or a baptism, may be found in Austin ([1955]1962). Their reconstruction as DBS inferences is analogous to 10.4.4 (perspective of the authorized official) and 10.5.1 (perspective of the engaged couple.)

The ninth class of inferences copies noun proplets resonating in memory as shadows by address to the now front:

10.6.9 Inference for shadowing

49. Inference for noun shadowing (6.6.2)

Shadowing by address allows stored content to participate in current processing without touching the data in the agent's memory.

The tenth class of inferences summarizes repeated occurrences of an episodic content into a generic content:

10.6.10 Generic Inference

50. Inference deriving generic STAR-0 content (12.3.1)

The ten classes of DBS inferences are based on concrete empirical tasks such as the reconstruction of some classical syllogisms, the changing perspective on a content's origin in the speak and the hear mode, significant observations such as no gas–no go, and nonliteral uses. In hindsight, the distinctions between the ten classes may be summarized in a more abstract way as the following three.

The first distinction is between obligatory and optional applications, which is well-known in linguistics, but hasn't been used in substitution-driven sign-based logical inferencing. An example of an obligatory inference is the reversion of pronominal indexicals from the speak mode to the hear mode. An example of an optional inference is a figurative use in the speak mode.

A second distinction is between set-theoretical and non-set-theoretical inferences. Inferences based on set theory are exemplified by the classical syllogisms. Inferences without any set-theoretic aspect are summarizing, cause and effect, and counter measure, in the speak and the hear mode. Until now, the distinction has not been used because in symbolic logic (1.2.3) inferences without set theory can only be handled by the ad hoc definitions of the formal model.

A third distinction is between deductive and abductive inferencing, which goes back at least to Peirce. Until now it has not been used for characterizing a class of inferences which are applied deductively in the speak mode and abductively in the hear mode.

In summary, DBS inferencing employs conceptual constructs of formal semantics, theoretical linguistics, philosophy of language, and cognitive psychology by using some of their well-studied and well-known notions, but applies them in new ways. The new applications are unified methodologically in the concomitant design of software constructs which are presented in the standardized DBS format of a declarative specification.

Part III

Data-Driven Behavior Control

11. Interface Component

Part III presents a standardized integrated software architecture of computational cognition in DBS. As a top down design, it unifies earlier bottom up reconstructions of phenomena treated traditionally as separate. The architecture aims at (a) accommodating all the interfaces, components, functional flows, and operations required by a computational cognition, (b) avoiding the artifacts (15.3.3, 15.3.5), gaps (15.3.7), puzzles (FoCL Sect. 22.2), and paradoxes (FoCL Sects. 8.5, 19.5) of the various sign-based systems, and (c) supporting incremental long-term upscaling which preserves the conceptual transparency and computational efficiency of the initial design.

Sect. 11.1 outlines the main parts of DBS cognition, called the (i) interface, (ii) memory, and (iii) operation components.[1] The sections remaining in this chapter are dedicated to (i) the interface component. Sect. 11.2 presents the sensory media of speech, writing, Braille, and signing, each with its dual modalities of recognition and action. Sect. 11.3 describes nonlanguage recognition as concept types matching raw data provided by sensors, and nonlanguage action as actuators producing raw data from concept tokens. Sects. 11.4 and 11.5 present the corresponding mechanisms for surface recognition (hear mode) and production (speak mode). Sect. 11.6 concludes with behavior in which recognition controls action and action controls recognition.

11.1 Three Main Components of Cognition

The (i) interface component manages the interaction between the cognitive agent and its environment, which consists of raw data. In practice, the computational processing of raw data as input and output relies on the natural and the engineering sciences. The interface component is the theoretical and practical foundation of a grounded semantics as part of the humanities.

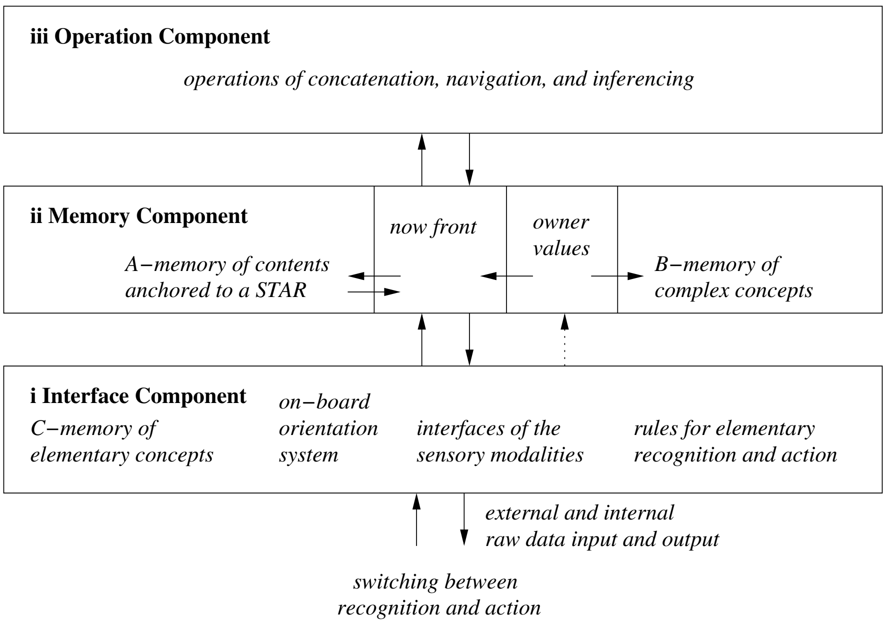
[1] The components correspond roughly to those of a v. Neumann machine (vNm, v. Neumann 1945): the (i) interface component corresponds to the vNm input-output device, the (ii) memory (database) component corresponds to the vNm memory, and the (iii) operation component performs functions of the vNm arithmetic-logic unit.

The (ii) memory component is the control center of cognition (in addition to its traditional use for storage and retrieval of the analyzed data provided by the interface and the operation components.) Defined as a content-addressable database, it has a two-dimensional schema. Horizontal token lines contain proplets with the same core value in the order of their arrival (2.3.2). Vertically, the token lines are in the alphabetical order induced by their shared core value.

The (iii) operation component provides (1) concatenation in recognition, and (2) navigation and inferencing in action, for language and nonlanguage content alike. The application of operations is data-driven and based on a strictly time-linear derivation order. An operation is activated by a content proplet matching its trigger pattern; the variables in the input pattern are bound to the corresponding input constants, enabling the operation output.

The component structure may be shown graphically as follows:

11.1.1 Two-dimensional layout of DBS cognition components

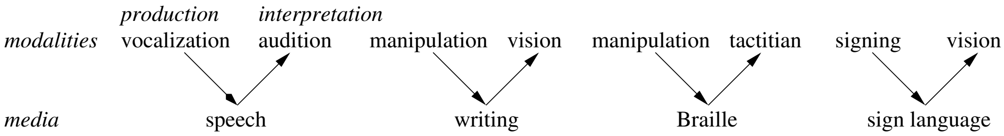


The cognitive arena in which content is processed is called the *now front*. It gets proplets (a) from the interface component (aided by the owners) and (b) from A-memory (shadowed by address, 6.6.2, 6.6.4). For processing, the now front provides proplets as input to (iii) the operations, which replace the input with their output (e.g. 2.2.2, 7.5.2). As the now front is cleared in regular intervals by moving into fresh memory space, the processed proplets are left behind in A-memory like sediment (loom-like clearance; 6.5.2, 6.5.3). Processing may also result in blueprints for action, which may be copied to the interface component for realization (subjunctive transfer, CLaTR Sect. 5.6).

11.2 Sensory vs. Processing Media with their Modalities

In natural language communication, there exist four sensory media, each of which has two sensory modalities.[2] For example, if the speaker chooses the medium of speech, the only sensory modality for production is vocalization (↘), which leaves the hearer no other option than using the sensory modality of audition (↗). This asymmetry of modalities holds also for the other sensory media of natural language, namely writing, Braille, and sign language.

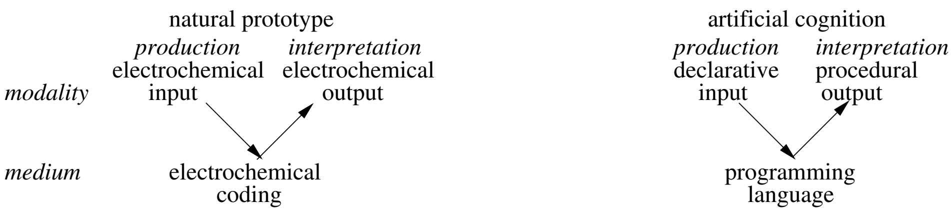
11.2.1 Sensory media and modalities in natural language



In terms of human evolution, the primary sensory medium is speech.

In addition to the sensory media there is the processing medium of cognition. For the natural prototype, neurology suggests an electrochemical processing medium, though much is still unknown.[3] In artificial DBS cognition, the processing medium is a programming language; its processing modalities are (i) the declarative specification of commands for interpretation by the computer and (ii) their procedural execution by the computer's electronic operations.

11.2.2 Processing media and their dual modalities



Utilizing a programming language as the processing medium of an artificial agent requires an interface component capable of efficiently mediating between raw data and an alphanumeric representation in recognition and action.[4]

---

[2] In the literature, the term modality has a multitude of uses, such as the temperature (Dodt and Zotterman 1952), the logical (Barcan Marcus 1961), and the epistemic (Kiefer 2018) modalities.

[3] For an early overview see Benson (1994), still unsurpassed.

[4] To permit a DBS robot to communicate in speech, a conversion from the sensory modalities of ma-

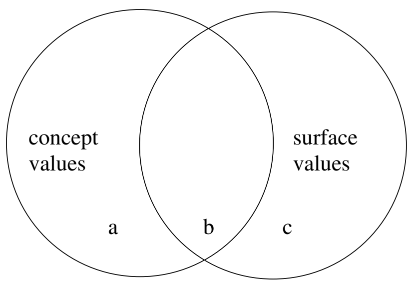
11.3 Mapping between Concepts and Raw Data

The interface component has four basic tasks, namely nonlanguage content (i) recognition and (ii) action (3.1.3), and language surface (iii) interpretation and (iv) production (3.1.1, 3.1.2). In immediate language reference (3.1.1), they may all be used simultaneously.

The four kinds of interaction between the talking robot's interface component and the raw data of its environment are based on four kinds of type/token pairs. One kind are the non-language concepts of (1) recognition and (2) action, which serve as values for many proplets' core and continuation attributes. The other are the surface values of language (3) interpretation and (4) production which serve as values for many proplets' sur attribute.

There is no one-to-one relation between the surfaces of a natural language and the concepts of nonlanguage cognition; instead they are in the following constellation:

11.3.1 CORRELATION OF CONCEPTS AND SURFACES IN A LANGUAGE



Proplets with a concept value but without a surface value, such as an unnamed color,[5] are in (a). Proplets which have a sur value in addition to a concept value are in (b). Proplets which have a sur value, but use a substitution variable (determiner), an operator (connective), or a pointer (indexical), are in (c).

Of the three elementary content kinds (i) concept, (ii) indexical, and (iii) name (1.5.4), only the concepts offer a choice between observation (recognition) and self-performance (action; Sect. 8.6). From a theory of science point

nipulation (writing) to articulation (speech synthesis) and from audition (hearing) to vision (speech recognition) must be implemented in order to allow the robot to use its alphanumeric processing medium (CLaTR 2.3.2) with the sensory medium of speech, and accordingly for Braille and signing.

In addition, there are the optional modality conversions performed by a hearer-speaker during interpretation as in converting vision into speech by reading aloud or speech into vision during dictation (NLC Sect. 2.2; CLaTR Sects. 2.2, 2.3.)

[5] For a recent paper on a language-dependent absence of distinctions in the visual modality as compared to English see Putten (2019). For the opposite direction, i.e. a more differentiated vocabulary as compared to English in a limited domain, see Boas (1911)

of view, recognition and action based on concept types matching raw data constitute a seminal interaction between the humanities on the one hand, and the natural and engineering sciences on the other.

The following rule shows nonlanguage recognition of a color as a mapping of raw data provided by the agent's vision sensor into an alphanumerical concept token (1.3.5).

11.3.2 Declarative specification for recognizing a color

$$
\begin{array}{cccc}
 & \textit{concept type} & & \textit{concept token} \\
 & \left[\begin{array}{l}
\text{place holder: blue} \\
\text{sensory modality: vision} \\
\text{semantic field: color} \\
\text{content kind: concept} \\
\text{wavelength: 450–495nm} \\
\text{frequency: 670–610 THz} \\
\text{samples: a, b, c, ...}
\end{array}\right] & \Rightarrow & \left[\begin{array}{l}
\text{place holder: blue} \\
\text{sensory modality: vision} \\
\text{semantic field: color} \\
\text{content kind: concept} \\
\text{wavelength: 470nm} \\
\text{frequency: 637 THz} \\
\text{samples: ...}
\end{array}\right] \\
 & \Uparrow & & \textit{analyzed output} \\
\textit{raw input} \;\Rightarrow & \begin{array}{l}
\text{sensory modality: vision} \\
\text{sensor values: 470nm} \\
\qquad\qquad\quad \text{637 THz}
\end{array} & &
\end{array}
$$

The concept type is supplied by C-memory. Retrieving the correct type, i.e. the one best matching the raw data at hand, is helped by (i) the data modality and (ii) narrowing the possibilities by using a subsumption hierarchy. In 11.3.2, the type recognizes the raw input data 470nm and 637 THz as the color blue because they fall into the type's wavelength interval of 450–495nm and the frequency interval of 670–610 THz. The analyzed output token results from replacing the wavelength and frequency intervals of the type with the raw data of the input.

The *place holder* value is used (i) in the column of owner values (2.3.3, 12.4.4) for accessing the associated token line at the now front, and as (ii) the core and continuation value in content proplets. The *sensory modality* value specifies the interface subcomponent providing the raw input data. The *semantic field* value (11.3.3) specifies the lexical class of the concept. The remaining features specify the concept itself.

A semantic field[6] (Trier 1928; Ullmann 1957; Andersen 1990) is a lexical classification which goes back at least to Wilhelm v. Humboldt.

11.3.3 Examples of semantic fields

- field: *color*
  elements: red, green, blue

[6] Brinton (2000) has called the semantic field relation a loose form of hyponymy (9.1.1), which in turn may be viewed as a containment relation between a set and its subsets.

- field: *water*
  elements: ocean, lake, river, pool
- field: *furniture*
  elements: chair, table, cupboard, shelf, bed

A concept type (11.3.2, 11.3.4, 11.4.1, 11.5.2) may specify more than one semantic field.

The action counterpart to the recognition 11.3.2 is the following rule. It takes a concept as input and renders raw data as output, as in a cuttlefish (Metasepia pfefferi) turning on the color blue:

11.3.4 DECLARATIVE SPECIFICATION FOR PRODUCING blue

concept type		*concept token*	
place holder: blue sensory modality: visual display semantic field: color content kind: concept wavelength: 450–495nm frequency: 670–610 THz samples: a, b, c, ...	⇒	place holder: blue sensory modality: visual display semantic field: color content kind: concept wavelength: 470nm frequency: 637 THz samples: ...	
analyzed input		⇓	
		sensory modality: vision actuator values: 470nm 637 THz	⇒ *raw output*

The type is adapted into a token by replacing the wavelength interval of 450–495nm and frequency interval of 670–610 THz with the specific values of 470nm and 637 THz. In the cuttlefish, these values are realized by natural actuators for color control (chromatophores) as raw data.

The number of concepts in an artificial agent's interface component may be large, their definition complex, and their implementation laborious. Yet after working out the basic functioning of a DBS robot's computational cognition in principle by using a small number of concepts in a codesigned but real environment, more concepts of the same kind may be added routinely. As examples compare the color concepts red, green, and blue[7]:

11.3.5 SIMILARITY AND DIFFERENCE BETWEEN COLOR CONCEPT TYPES

place holder: red	place holder: green	place holder: blue
sensory modality: vision	sensory modality: vision	sensory modality: vision
semantic field: color	semantic field: color	semantic field: color
content kind: concept	content kind: concept	content kind: concept
wavelength: 700-635 nm	wavelength:495-570 nm	wavelength: 490-450 nm
frequency: 430-480 THz	frequency: 526-606 THz	frequency: 610-670 THz
samples: a, b, c, ...	samples: a′, b′, c′, ...	samples: a″, b″, c″, ...

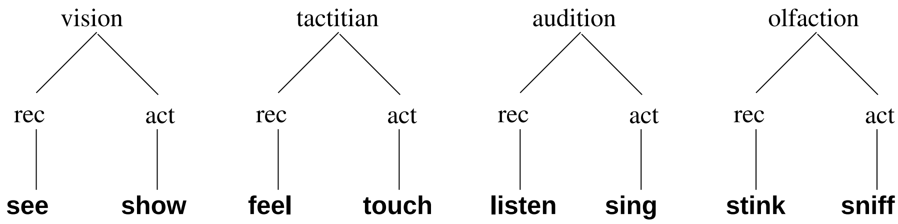
[7] Red, green, and blue are (i) disjunct and (ii) subsets of color. This set-theoretical structure is inherent in the color concepts, but it is neither the only nor the predominant aspect of their meaning.

Once the recognition (11.3.2) and action (11.3.4) side of these concepts is working as intended, more color concepts may be easily added as an efficient, transparent upscaling.

The same works for geometric forms: once the concepts of square (1.3.2) and rectangle work as intended, more two-dimensional forms, such as triangle, heptagon, hexagon, and rhombus, may be added routinely. After implementing the concept pick including the associated hand-eye coordination and the semantic relation of object\predicate (2.5.1, 2), the robot should be able to execute requests like Pick the blue square or Pick the green rectangle correctly from a set of items in its task environment.

There are concepts which are suitable for both observation (recognition) and self-performance (action). In natural language, such alternative uses of a given concept often have different surfaces:

11.3.6 SURFACES DISTINGUISHING RECOGNITION VS. ACTION USE



If the subject is pro1, an action concept may be used for action, otherwise for recognition, as illustrated by the following example:

11.3.7 ALTERNATIVE USES OF THE ACTION CONCEPT show

(a) Recognition: Mary show picture
(b) Action: pro1 show picture

Example (a) illustrates a STAR-0 content which resulted from recording a recognition, example (b) a STAR-0 content which resulted from recording a self-performed action.[8] The distribution of recognition concepts with or without an action counterpart and the supply of distinctive surfaces depends on the language and the underlying culture.

[8] Use of a concept in recognition vs. action is distinct from, but related to, its use as the core value in a noun, verb, or adj proplet (NLC Sect. 4.2; CLaTR 6.6.1–6.6.3). For example, the recognition concept red may serve as the core value of a noun as in Mary prefers this red (referent), a verb, as in the sun reddened (relation) the sky, or an adnominal adj, as in the red (property) dress.

11.4 Hear Mode: Surface Interpretation

The language counterpart to the recognition and action of nonlanguage concepts is the interpretation and production of language-dependent surfaces. On the one hand, the set of language surfaces does not cover all concepts, but covers function words and indexicals in addition (11.3.1). On the other hand, concept and surface recognition and action are similar insofar as they are based on a computational pattern matching between types and raw data.

As an example, consider the DBS robot's recognition of the surface wolves by matching raw visual input data with a surface type, which results in a surface token:

11.4.1 Letter types matching raw data in the vision modality

pattern: surface type		*output: surface token*
place holder: wolf		place holder: wolf
sensory modality: vision		sensory modality.: vision
semantic field: language surface	⇒	semantic field: language surface
content kind: roman letters		content kind: roman letters
shape types: w o l v e s		sensor values: shape tokens w% o% l% v% e% s%
samples: ...		samples: ...
⇑		
sensory modality: vision		
input: raw data		

The input consists of raw data which are provided by the agent's vison sensors and matched by the letters' shape types provided by the agent's memory.[9] The output replaces the shape types, here w o l v e s matching the raw data with the shape tokens, here w% o% l% v% e% s%, to record such accidental properties as the font, size, color, etc. in the sensory medium of print, and pronunciation, pitch, speed, loudness, etc. in the medium of speech. The function crucial for reconstructing the cycle of communication, however, is using the shape types for computational look-up of the lexical definition.

Raw data matching in different media may be temporarily replaced by the shortcut of typing letters directly into the computer's key board and displaying letters on the computer screen. This has made it possible to implement three basic approaches to morphology as running software (FoCL Chaps. 13-15).

The (i) full-form approach[10] is based on two steps: step 1 is the recognition of the surface; step 2 uses the recognized surface for lookup of a corresponding entry in a full-form lexicon.[11] The approach has the advantage that

[9] For ease of illustration, the letter shapes are represented by the letters themselves, e.g. e (type) and e% (token).

[10] The full-form approach was used in the primitive lexical lookup of the NEWCAT software written in Lisp and published in 1986.

a compositional analysis of complex word forms is avoided. The disadvantage from a linguistic point of view is a failure to capture morphological regularities. From a software point of view, the disadvantage is a much greater space consumption than that of a rule-based approach (FoCL 14.1.6). Also, adding new entries requires needless work.

The (ii) morpheme approach (Koskenniemi 1983) is based on four steps. Step 1 segments the surface into its allomorphs. Step 2 reduces the allomorphs to their morphemes. Step 3 is lookup of the morpheme definitions in a morpheme lexicon. Step 4 is morpheme concatenation by rules:

11.4.2 THE FOUR STEPS OF THE MORPHEME APPROACH

1. surface segmentation: wolv/es (allomorphs)

2. morpheme reduction: wolf pm

3. morpheme lookup:

$$\left[\begin{array}{l}\text{sur:}\\ \text{noun: wolf}\\ \text{cat: n}\\ \text{sem:}\\ \text{fnc:}\\ \dots\\ \text{prn:}\end{array}\right] \quad \left[\begin{array}{l}\text{sur:}\\ \text{noun: pm}\\ \text{cat: p}\\ \text{sem:}\\ \text{fnc:}\\ \dots\\ \text{prn:}\end{array}\right]$$

4. morpheme concatenation:

$$\left[\begin{array}{l}\text{sur:}\\ \text{noun: wolf+pm}\\ \text{cat: pn}\\ \text{sem:}\\ \text{fnc:}\\ \dots\\ \text{prn:}\end{array}\right]$$

From a linguistic point of view, the concatenation of morphemes rather than allomorphs is reminiscent of using a deep structure, as in generative grammar.

The other rule-based approach is the (iii) allomorph[12] approach (Handl et al. 2009), which is based on three steps. Step 1 segments the surface into its allomorphs (similar to the morpheme approach). Step 2 is look-up of the allomorph definitions in an allomorph lexicon which was derived from a morpheme lexicon before run time (FoCL 13.5.3).[13] Step 3 is the concatenation of the analyzed allomorphs into the complete analyzed word form.

[11] For example, the letter sequence t a b l e is recognized as a surface of English with an associated lexical entry, while the inverse sequence e l b a t is not.

[12] A class of allomorphs contains language-dependent surface variants of the same elementary concept, whereby the class is called the morpheme (FoCL 13.2.4). For example, {swim, swimm, swam, swum} are allomorphs (allographs) of the morpheme swim, {wolf, wolv} are allomorphs of the morpheme wolf, and {s, es, en, ren, ...} are allomorphs of the "plural morpheme."

[13] During upscaling, this step is needed only once in a while, namely after correction and/or extension of the analyzed data.

11.4.3 THE THREE STEPS OF THE ALLOMORPH APPROACH

1. surface segmentation:	wolv/es (allomorphs)
2. allomorph lookup	$\begin{bmatrix}\text{sur: wolv}\\ \text{noun: wolf}\\ \text{cat: pn sr}\end{bmatrix} \begin{bmatrix}\text{sur: es}\\ \text{suffix: plural}\\ \text{cat: pl}\end{bmatrix}$
3. allomorph concatenation	$\begin{bmatrix}\text{sur: wolv+es}\\ \text{noun: wolf}\\ \text{cat: pn}\\ \text{sem:}\\ \text{fnc:}\\ \dots\\ \text{prn:}\end{bmatrix}$

Compared to the full-form and the morpheme approach, precomputed allomorphs (12.5.3) result in the shortest run time and the simplest concatenation.[14]

11.5 Speak Mode: Surface Production

Automatic word form production is the language counterpart to nonlanguage action (11.3.4). While word form recognition is a mapping from a surface to a proplet, production is a mapping from a proplet to a surface.

11.5.1 COMPARING WORD FORM RECOGNITION WITH PRODUCTION

	word form recognition:		*word form production*:
input	raw surface	*output*	raw surface
transition 1	$\Downarrow$	*transition 2*	$\Uparrow$
	recognized surface		analyzed surface
transition 2	$\Downarrow$	*transition 1*	$\Uparrow$
output	lexical proplet	*input*	connected proplet

Word form recognition and production are alike in that each requires two transitions, but differ in the direction. The output of word form recognition is isolated (lexical) proplets. The input to the speak mode, in contrast, are proplets which are connected into content by the semantic relations of structure .

Consider the realization of a German surface as a letter sequence. The input is a connected proplet with the core value wolf and the sem values pl def. The output is the German surface d i e W ö l f e on the computer screen.

[14] The software for the allomorph approach was developed by G. Schüller (1994), O. Lorenz (1996), B. Beutel (1997), A. Kycia (2004), and J. Handl (2008, 2012). The development of the DBS software in Lisp, Java, and C up to the year 2011 is summarized by Handl (2012) in chapters 6 and 7. For a list of the natural languages analyzed in DBS and the researchers who did the work see CLaTR 3.5.3.

11.5.2 From connected proplet to raw surface output

input: connected proplet

$$\begin{bmatrix}\text{sur:}\\ \text{noun: wolf}\\ \text{cat: pnp}\\ \text{sem: pl def}\\ \text{fnc: save}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 67}\end{bmatrix}$$

$$\Downarrow$$

output: analyzed surface

$$\begin{bmatrix}\text{place holder: wolf}\\ \text{sensory modality: manipulation}\\ \text{semantic field: language surface}\\ \text{content kind: Roman letters}\\ \text{actuator values: type shapes d i e W ö l v e}\\ \text{samples: ...}\end{bmatrix} \Rightarrow \text{\%d \%i \%e \%W \%ö \%l \%v \%e}$$

raw output

The surface type of German d i e W ö l f e is based on a list which provides German surface allomorphs for the relevant proplet values. For example, in German the feature [sem: pl def] is realized as the language-dependent determiner die, the language-independent core value wolf is realized as the corresponding stem Wölf (allomorph), and the plural morpheme pl as the stem suffix -e. The most detailed DBS analysis of surface realization so far may be found in NLC Sects. 12.4 (verbs), 12.5 (nouns) and 12.6 (adjs) for English.

11.6 Moment by Moment Monitoring

Depending on their semantic field value, the proplets of a content resulting from current recognition are directed to one of two storage locations: (i) the now front of A-memory or (ii) a STAR-0 slot. For example, a color recognition is represented as a current content proplet and stored at the now front in the token line of its core value (2.3.3). A current datetime value, in contrast, is stored in the T slot of the current STAR-0 proplet (6.3.4).

11.6.1 STAR-0 template of the on-board orientation system

STAR–0 Template

	sensors		
Space:	vision	←	
Time:	audition		
Agent: lucy		*raw data*	
Recipient:	*actuators*		
3rd:	manipulation	→	
prn:	locomotion		

The agent lucy systematically provides the A-value to all of her STAR-0s; as a cognition-internal value, no sensory modality is involved. The S, T, R, and 3rd values, in contrast, take values from different sensors and activators. For example, the T value may be provided by audition (radio) or vision (wrist watch).

While providing a STAR-0 with proper values is primarily a matter of recognition, it may also require action. For example, if the agent's on-board orientation registers that the current Time value has not yet been supplied, the action of, e.g. looking at one's watch (sensor controlling actuator), is initiated. The steps of alternating between recognition and action in such a case may be summarized as follows:

11.6.2 Steps of supplying a missing STAR-0 value

1. STAR-0 slot reports missing value (internal recognition)
2. on-board orientation searches for appropriate value (external action)
3. on-board orientation finds appropriate value (external recognition)
4. value is moved into appropriate slot of the STAR-0 (internal action)

Once a STAR-0 is complete, it is supplied to the now front. There it (a) connects to the other proplets by means of a shared prn value, (b) provides indexicals with values to point at, and (c) anchors the content in Space, Time, and, if appropriate, Recipient (partner of discourse) and 3rd (other).

Regardless of whether a recognition value is stored in a STAR-0 slot or at the now front, recognition values may often be gleaned from short lists of defaults, such as family members or other familiar persons, the neighbor's car, but also locations and times of day, week, or year. In these cases, recognition may be simplified into checking a few characteristic traits based on feature hierarchies.

In a new acquaintance or an unfamiliar object, however, the recognition sequence may have to proceed from scratch by matching concept types onto raw data. For future use, a successful result is added to the appropriate default list in accordance with the ordering algorithm of choice, for example, frequency using a hash.

An algorithm for recognizing unfamiliar objects in vision may use sensors controlling actuators and actuators controlling sensors as follows:[15] it starts with the recognition of something seen before, continues by searching for related contents (e.g. experiences or something learned) from memory (coactivation, Sect. 3.2), and then guides the sensors to obtain additional relevant recognition in the field of vision.

A concrete example is a physician examining a rash. As a procedure of *looking closer*, it requires the coactivation of stored content (3.2.1, 3.2.3), using such owners as rash, location, appearance, activity, and medication. For systematically excluding possibilities, Parallel Application (PAP, 3.6.4) may be used:

11.6.3 Example of a Parallel Application

pro1 see rash $\Rightarrow$ pro1 study appearance.
pro1 see rash $\Rightarrow$ pro1 look for locations.
pro1 see rash $\Rightarrow$ pro1 ask patient about recent activities
...

The cycle ends when the finite supply of new consequents for the same antecedent has been exhausted; the number of results is limited by the amount of matching data provided by memory (3.6.5). If an output is produced, as when the patient turns out to take a certain medication, the process may switch to a routine of questioning and advising which resembles sequential activation (SAP, 3.6.2).

[15] "Memory-based pattern completion" (Hausser 2005) based on Biederman's (1987) geons. Thanks to Professor Brian MacWhinney for bringing Biederman's work to the author's attention.

Terminological Remark

When Left-Associative Grammar (LA grammar, LAG) was first programmed as the NEWCAT parser at CSLI Stanford in 1984–1986, it resembled the sign-based main stream approaches in that there was no agent, no distinction between recognition and action in general, and no distinction between the hear and the speak mode in particular.[16] LA grammar was defined as a traditional algorithm with a set of start states, a set of rules, and a set of final states.

A rule of this algorithm consisted of a pattern for a sentence start, a next word, and the result, plus a rule package allowing and limiting the rules to be tried next on the new next word and the current sentence start. The overall purpose of LA-grammar at this stage was to demonstrate that computing possible continuations (time-linear concatenation) instead of possible substitutions (partially ordered replacements) is suitable for an efficient, transparent syntactic-semantic analysis of as many constructions in modern grammar theory for English and German as possible.

With the subsequent development of the proplet data structure and A-memory's content-addressable database schema, the role of the rule packages to restrict current input data became obsolete and was replaced by (1) the now front constraining the input supply and (2) the data-driven application of operations, using the input supply as the automatic trigger. To make this development explicit, the terminology was changed from the "rules" of the original LA grammar to the "operations" of DBS.

This freed the term "rule" for another use, namely for mappings between alphanumerical structures and raw data in the interface component (11.3.2, 11.3.4, 11.4.1, 11.5.2). At the same time, the term "operation" was limited to alphanumerical structures without any involvement of raw data (i.e. purely cognition-internal), such as left-associative concatenation (2.2.1) in the hear mode, left-associative navigation (2.6.2) and inferencing (3.5.1) in the think mode, and the production of surface types in the think-speak mode.

[16] The development from a sign-based to an agent-based system is described in Hausser (2015).

12. Memory Component

The second main component of computational DBS cognition is the agent's onboard memory.[1] Implemented as a content-addressable database, it provides storage and retrieval, but also has a now front for limiting potential input to data-driven operations and for coactivating resonating content in storage.

Sect. 12.1 describes the division of the agent's on-board database into the main section of A-memory and the auxiliary sections of B- and C-memory. Sect. 12.2 summarizes the interpretation of indexicals and names. Sect. 12.3 compares generic content with episodic content and outlines the inference for deriving the former from repetitions of the latter. Sect. 12.4 shows the inherent relation between recognition and storage in the hear mode. Sect. 12.5 illustrates a standard computational retrieval method in the medium of writing, namely *string search* (Knuth et al. 1977). Sect. 12.6 shows the inherent relation between action and retrieval of a blueprint for action in the speak mode.

12.1 The Memory Sections of DBS Cognition

The on-board memory of DBS cognition has three sections: (a) A-memory for storage and retrieval of episodic and generic contents **a**nchored to the on-board orientation system (STAR), (b) B-memory for complex concepts such as recognition and action routines without a STAR (**b**are), and (c) C-memory for elementary **c**oncepts. B- and C-memory provide the declarative specifications and operational implementations for the place holder values in A-memory. A- and B-memory are located in the memory component, and C-memory in the interface component (11.1.1).

[1] In the neighboring science of cognitive psychology, work on human memory has centered on such questions as to whether memory strength is being continuously variable (multi-store model by Atkinson and Shiffrin 1968), or encoded and processed at different levels in the brain (levels-of-processing model by Craik and Lockhardt 1972). Other models are the single channel theory (Broadbent 1958), the capacity theory (Kahneman 1973), and the multiple resource theory (Navon and Gopher 1979). The computational counterpart are subsymbolic neural networks (McCulloch and Pitts 1943, Smolensky 1991, Hinton 1992, Thagard 2006, and others) which model learning by building inhibitory and excitatory connections in multilevel networks (connectionism).

The proplets in A-memory are placed to the left and the definitions in B-memory to the right of the column of owners (11.1.1). This enables A- and B-memory to share the owners' trie structure for efficient storage and retrieval:

12.1.1 Vertical and horizontal aspects of A- and B-memory

(i) A-*memory*	(ii) *now front*	(iii) *owners*	(iv) B-*memory*
[verb: cook arg: man potatoe prn: 27]		**cook**	definition of cook (12.1.2)
...			
[noun: man fnc: cook prn: 27]		**man**[2]	
...			
[noun: potatoe fnc: cook prn: 27]		**potatoe**	

Horizontally, the memory component consists of token lines. A token line is composed of the (i) member proplets in A-memory, a (ii) free slot belonging to the now front column, the (iii) owner, and (iv), if appropriate, an associated entry in B-memory. The operational implementation of complex concepts in B-memory are an important factor in maintaining the agent's balance. Examples are not spilling when drinking from a cup, not stumbling when stair-climbing, and not falling to the floor when sitting down.

As an example of a complex concept in B-memory consider the following definition of cook as a procedure:

12.1.2 Definition of cook in B-memory

> cook (key word)
> If $\alpha\ \varepsilon$ {person, man, woman, ...}, $\beta\ \varepsilon$ {potatoes, rice, nudels, ...}, and α cook β, then α put β into pot, fill up with water, turn up heat on stove, bring water to boil, and if α = potatoes cook for 20 minutes.

The road to a consistent systematics of a B-definition like 12.1.2 is an embedding into the appropriate semantic field. For example, the field prepare deals not only with the preparation of food, but also of lecture, trip, surgery, etc., in a subsumption hierarchy. At the lowest level, practical and even cultural questions of whether or not, for example, using a rice cooker should be integrated into the definition, is a matter of the robot's intended purpose.

---

[2] The elementary concepts man and potatoe are defined in C-memory (interface component).

Like the elementary concepts in C-memory, the definitions of complex concepts in B-memory rely on the implementation in the interface component, not only regarding the sensory modalities of recognition and action in general, but in detailed concrete procedures for individual concepts. This has the theoretical consequence that a definition like 12.1.2 does not suffer from circular paraphrasing (Sect. 15.3): even if some concepts are defined in terms of other concepts, there is a level of basic recognition and action in which concepts interact with raw data directly (11.3.2, 11.3.4). These elementary concepts are used to supply the place holder values with a foundation of concrete declarative and operational definitions (1.6.3).

The elementary concepts of C-memory and the complex concepts in B-memory do not have their own STAR, but borrow the STAR of the A-contents in which they serve. Because B- and C-contents are not monitored outside their support role in A-memory, they do not require a writable memory. The contents in A-, B-, and C-memory reflect the agent's ecological niche and are used in nonlanguage and language cognition alike.

12.2 Interpretation of Indexicals and Names

Of the three content kinds concept, indexical, and name (1.5.4), only concept and name (with its core value, i.e. the named referent) participate in subactivating resonating content in A-memory by using the access structure of the owners and the arrival order in the token lines. Indexicals, in contrast, do not relate back to an initial or a named referent, but point forward at associated values in the current STAR. Thereby, the interpretation of the subject in pro1 feel hungry, for example, varies with the A value of the STAR:

12.2.1 DIFFERENT INDEXICAL INTERPRETATIONS OF SAME CONTENT

$$
\text{(i)} \quad
\begin{bmatrix} \text{noun: pro1} \\ \text{fnc: feel} \\ \text{prn: 22} \end{bmatrix}
\begin{bmatrix} \text{verb: feel} \\ \text{sem: pres ind} \\ \text{arg: pro1} \\ \text{mdr: hungry} \\ \text{prn: 22} \end{bmatrix}
\begin{bmatrix} \text{adj: hungry} \\ \text{mdd: feel} \\ \text{prn: 22} \end{bmatrix}
\begin{bmatrix} \text{S: kitchen} \\ \text{T: 20080915T155300} \\ \text{A: sylvester} \\ \text{R:} \\ \text{3rd:} \\ \text{prn: 22} \end{bmatrix}
$$

$$
\text{(ii)} \quad
\begin{bmatrix} \text{noun: pro1} \\ \text{fnc: feel} \\ \text{prn: 39} \end{bmatrix}
\begin{bmatrix} \text{verb: feel} \\ \text{sem: pres ind} \\ \text{arg: pro1} \\ \text{mdr: hungry} \\ \text{prn: 39} \end{bmatrix}
\begin{bmatrix} \text{adj: hungry} \\ \text{mdd: feel} \\ \text{prn: 39} \end{bmatrix}
\begin{bmatrix} \text{S: yard} \\ \text{T: 20080915T093200} \\ \text{A: hector} \\ \text{R:} \\ \text{3rd:} \\ \text{prn: 39} \end{bmatrix}
$$

Apart from the prn values, the content proplets are identical. Pragmatically, however, pro1 points at the A-value sylvester in (i), while the same indexical points at the A-value hector in (ii).

Of the personal pronouns, pro3 is special compared to pro1 and pro2 because pro3 has not only an indexical, but also a coreferential interpretation in extrapropositional relations. For example, in Fido finds a bone. He hides it., the pronouns he and it may refer not only indexically, but also coreferentially back to Fido and bone:

12.2.2 Coreferential pronoun interpretation in coordination

$$
\begin{bmatrix}\text{sur: fido}\\ \text{noun: [dog 25]}\\ \text{fnc: find}\\ \text{prn: 81}\end{bmatrix}
\begin{bmatrix}\text{verb: find}\\ \text{arg: [dog 25] bone}\\ \text{nc: (hide 82)}\\ \text{prn: 81}\end{bmatrix}
\begin{bmatrix}\text{noun: bone}\\ \text{verb: find}\\ \text{prn: 81}\end{bmatrix}
\begin{bmatrix}\text{S: yard}\\ \text{T: ...T101630}\\ \text{A: john}\\ \text{R:}\\ \text{3rd:}\\ \text{prn: 81}\end{bmatrix}
$$

$$
\begin{bmatrix}\text{sur: fido}\\ \text{noun: [dog 25]}\\ \text{fnc: hide}\\ \text{prn: 82}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{verb: hide}\\ \text{arg: [dog 25] (bone 81)}\\ \text{prn: 82}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{noun: (bone 81)}\\ \text{fnc: hide}\\ \text{prn: 82}\end{bmatrix}
\begin{bmatrix}\text{S: yard}\\ \text{T: ...T101635}\\ \text{A: john}\\ \text{R:}\\ \text{3rd:}\\ \text{prn: 82}\end{bmatrix}
$$

The two propositions are connected by the feature [nc: (hide 82)] of *find* (extrapropositional coordination). Even though there are no indexicals pointing at STAR values, the two addresses [dog 25] (named referent) and (bone 81) in noun slot of the second proposition may be realized in English as the pronouns he and it in the coreferential (rather than the indexical) use.[3]

The other kind of extrapropositional relations in natural language besides coordination is functor-argument, as in When Fido finds a bone, he hides it. Representations of a coreferential noun (i) in extrapropositional coordination and (ii) in extrapropositional functor-argument as a pronoun are alike in that they require grammatical compatibility.[4]

However, even if a noun and a potentially coreferent 3rd person pronoun are grammatically compatible within an extrapropositional functor-argument, coreference may still be prevented by the Langacker-Ross constraint.[5] The constraint excludes third person pronouns in a lower clause from coreference with a potential postcedent in a higher clause. For example, in She thought

[3] A non-pronominal alternative would be Fido finds a bone. Fido hides the bone., but the pronominal variant is preferred.

[4] For example, the subjects in The woman slept. He snored. (coordination) or When the woman slept, he snored. (functor-argument) can not have a coreferential interpretation because the grammatical genders of woman and he do not agree.

that Lucy was pregnant, she and Lucy can not have a coreferential interpretation. In Lucy thought that she was pregnant, in contrast, Lucy and she may refer to the same person, i.e. be coreferential in addition to an indexical interpretation (which is always possible, excepting reflexive pronouns; Lees 1960, Lees and Klima 1963).

Besides concepts and indexicals, there are the names[6] as the third elementary Content kind in natural language (1.5.4). Name proplets are special in that (a) the core value is an address referring to the *named referent* which resulted in an implicit[7] or explicit act of baptism and (b) the sur slot contains a marker representing the name (CTGR Sect. 8[8]). The named referent, e.g. [dog x], is used in the hear mode for surface interpretation, while the marker, e.g. fido, in the sur slot is used in the speak mode for surface realization.

12.3 Episodic vs. Generic Contents

The agent's A-memory contains *episodic* and *generic* contents. They use the same proplet shells (NLC Sect. 4.2; CLaTR Sect. 6.6), are connected by the same semantic relations of structure, and are anchored to STAR proplets. They differ in that the proplets of an episodic content are anchored to a full STAR proplet, while the proplets of a generic content are anchored to a timeless STAR proplet. Timeless STAR proplets use the T value ∞ instead of datetime, and add gen(eric) to the prn value, e.g. gen20 (12.3.1).

When an agent in the speak mode reports an ongoing event, as in I am walking on the mobile phone, this is an episodic content which is expressed here by using the present progressive.[9] A present indicative, if not used as a 'literary

---

[5] Langacker (1969), Ross (1969). See CLaTR Sects. 11.3–11.5 for linguistic analysis and explanation.

[6] For Russell (1905) a name is *proper* if it satisfies his celebrated uniqueness condition, formulated in Predicate Calculus as $\exists x[f(x) \wedge g(x) \wedge \forall y[f(y) \rightarrow y = x]]$, with f for King of France and g is bald. DBS, in contrast, handles plural names like the Millers via the cat and sem values. Because names need not have unique referents, the epithet proper is obsolete in DBS.

[7] As when the agent witnesses, for example, an unknown dog being called by name.

[8] In the CASM (BioMed Central, Springer) edition of CTGR, all numbering of sections and examples had to be removed. For example, instead of referring to Sect. 8, the section heading had to be used, here '"Reference by Baptism (Name)" section.' In this awkward manner, the electronic publisher eliminated the possibility of automatic cross-referencing (`\hyperref`). For the original paper see `lagrammar.net`.

[9] English frequently uses the present progressive for episodic and the present indicative for generic content, coded in DBS as the verb features [sem: pres ind] vs. [sem: pres prog]. For example, I buy the rolls (habitually) is generic while I am buying rolls (as when answering "what are you doing?") is episodic.

German does not code the distinction analytically in the morphology, but synthetically in the syntax by means of adverbs like gerade and immer. For example, Ich gehe gerade die Brötchen holen (I walk straight the rolls fetch). is episodic, while Ich gehe immer die Brötchen holen (I present,' may characterize a generic content, like Whales are mammals (reference to a general kind), Mary takes a nap in the afternoon (perceived regularity), or The earth is flat (presumed knowledge).

Alternative to attempts at characterizing generic contents in terms of truth conditions (Carlson 1982; Asher and Pelletier 2012; ter Meulen 2012), DBS characterizes the difference between episodic and generic content as a difference in perspective. It is treated by an inference which takes a sequence of repeating contents as input and derives a corresponding single content with a timeless STAR-0 proplet as output. As an example, consider the following outline of deriving a generic content based on three instances of the episodic proposition I am walking to school as input and producing the generic proposition I walk to school (habitually) as output (only content level shown):

12.3.1 Deriving generic STAR-0 content by inference

$$\begin{bmatrix}\text{noun: pro1}\\ \text{fnc: walk}\\ \text{prn: 12}\end{bmatrix} \begin{bmatrix}\text{verb: walk}\\ \text{sem: pres prog}\\ \text{arg: pro1}\\ \text{mdr: school}\\ \text{prn: 12}\end{bmatrix} \begin{bmatrix}\text{noun: school}\\ \text{sem: } \textit{to}\\ \text{mdd: walk}\\ \text{prn 12}\end{bmatrix} \begin{bmatrix}\text{S: austin}\\ \text{T: Monday}\\ \text{A: percy}\\ \text{R:}\\ \text{prn: 12}\end{bmatrix} \begin{bmatrix}\text{noun: pro1}\\ \text{fnc: walk}\\ \text{prn: 16}\end{bmatrix} \begin{bmatrix}\text{verb: walk}\\ \text{sem: pres prog}\\ \text{arg: pro1}\\ \text{mdr: school}\\ \text{prn: 16}\end{bmatrix}$$

$$\begin{bmatrix}\text{noun: school}\\ \text{sem: } \textit{to}\\ \text{mdd: walk}\\ \text{prn 16}\end{bmatrix} \begin{bmatrix}\text{S: austin}\\ \text{T: Tuesday}\\ \text{A: percy}\\ \text{R:}\\ \text{prn: 16}\end{bmatrix} \begin{bmatrix}\text{noun: pro1}\\ \text{fnc: walk}\\ \text{prn: 20}\end{bmatrix} \begin{bmatrix}\text{verb: walk}\\ \text{sem: pres prog}\\ \text{arg: pro1}\\ \text{mdr: school}\\ \text{prn: 20}\end{bmatrix} \begin{bmatrix}\text{noun: school}\\ \text{sem: } \textit{to}\\ \text{mdd: walk}\\ \text{prn 20}\end{bmatrix} \begin{bmatrix}\text{S: austin}\\ \text{T: Wednesday}\\ \text{A: percy}\\ \text{R:}\\ \text{prn: 20}\end{bmatrix} \ldots$$

$$\Rightarrow \begin{bmatrix}\text{noun: pro1}\\ \text{fnc: walk}\\ \text{prn: gen20}\end{bmatrix} \begin{bmatrix}\text{verb: walk}\\ \text{sem: pres ind}\\ \text{arg: pro1}\\ \text{mdr: school}\\ \text{prn: gen20}\end{bmatrix} \begin{bmatrix}\text{noun: school}\\ \text{sem: } \textit{to}\\ \text{mdd: walk}\\ \text{prn: gen20}\end{bmatrix} \begin{bmatrix}\text{S: austin}\\ \text{T: } \infty\\ \text{A: percy}\\ \text{R:}\\ \text{prn: gen20}\end{bmatrix}$$

The grammatical difference between the three input propositions and the resulting output proposition is the feature [sem: pres prog] in the verb proplets of the input and the corresponding feature [sem: pres ind] in the verb proplet of the output.

The syntactic-semantic analysis of generic constructions in English is a wide field. For example, repetitions of episodic Fido is barking. may be turned into the generic variant Fido barked. (*past* indicative)[10] if Fido is deceased

walk always the rolls fetch) is generic. Both variants use the [sem: pres ind] form of the verb.

[10] The cat values of the past indicative differ in English and German. In English, there is only one form, for example, barked with the cat value n′ for the subject. In German, in contrast, there are four forms: bellte, belltest, belltet, and bellten with the distinctive cat values ns13′, ns2′, np2′, and np13′, respectively. The DBS hear mode preserves these values by #-canceling (instead of deletion, as in Categorial Grammar) because they may be needed for a surface realization in the narrative speak mode. These kinds of differences between the content representations of natural languages are a reason why content in DBS is only quasi language-independent.

and was an eager barker. There are also generic gapping constructions such as Chuck owns, Barbara cleans, and Alice drives the bus. (TExer Sect. 5.4).

12.4 Word Form Recognition and Morphology: Storage

The first step of the hear mode is automatic word form recognition. In DBS, this step relies in part on traditional Latin-style morphology with its conjugation of verbs, declination of nouns, and degrees of comparison in adjs.

12.4.1 CONJUGATION OF THE VERB know (distinctive categorization)

sur: know	sur: knows	sur: knew	sur: known	sur: knowing
verb: know	verb: know	verb: know	verb: know	verb: know
cat: n-s3′ a′ v	cat: ns3′ a′ v	cat: n′ a′ v	cat: a′ hv	cat: a′ be
sem: pres ind	sem: pres ind	sem: past ind	sem: perf	sem: prog
arg:	arg:	arg:	arg:	arg:
mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:
prn:	prn:	prn:	prn:	prn:

The variations of the sur values are by suffixation (know/s, know/n, know/ing) and by allomorphy (knew), and are accompanied by differences in all cat and in almost all sem values. The finite forms are distinguished by the cat values n-s3 for nominative without third person singular, ns3 for nominative third person singular, n for nominative, -g for for nob genitive, and g for genitive. The nonfinite forms lack a nominative valency and are distinguished by the cat values for the auxiliaries hv for 'have' and be for 'be.'

The four forms of some English common noun declinations are distinguished by the cat values sn for singular noun, pn for plural noun, -g for without genitive, and g for genitive.

12.4.2 DECLINATION OF THE NOUN woman (distinctive categorization)

sur: woman	sur: woman's	sur: women	sur: women's
noun: woman	noun: woman	noun: woman	noun: woman
cat: sn -g	cat: sn g	cat: pn -g	cat: pn g
sem: sg f	sem: sg f	sem: [11]	sem:
fnc:	fnc:	fnc:	fnc:
dots	...	...	...
prn:	prn:	prn:	prn:

The specification of grammatical gender in the singular as a sem value, here f for female, is needed for the coreference with the pronouns she and her.

[11] No grammatical gender distinction in common noun plurals of English (or German).

The degrees of comparison are distinguished by the sem values pad for positive adj, cad for comparative adj, and sad for superlative adj.

12.4.3 Degrees of comparison in the adv tall (distinctive categoriz.)

$$
\begin{bmatrix} \text{sur: tall} \\ \text{noun: tall} \\ \text{cat: adnv} \\ \text{sem: pad} \\ \text{mdr:} \\ \text{mdd:} \\ \dots \\ \text{prn:} \end{bmatrix}
\begin{bmatrix} \text{sur: taller} \\ \text{noun: tall} \\ \text{cat: adnv} \\ \text{sem: cad} \\ \text{mdd:} \\ \dots \\ \text{prn:} \end{bmatrix}
\begin{bmatrix} \text{sur: tallest} \\ \text{noun: tall} \\ \text{cat: adnv} \\ \text{sem: sad} \\ \text{mdd:} \\ \dots \\ \text{prn:} \end{bmatrix}
$$

All forms in each of the three inflections use the same core value, which functions like a traditional *root*. From a computational morphology point of view, however, the core value has the additional function of being the *key* for storage and retrieval, both within a natural language and between languages. Other difference are the (i) distinctive categorization (FoCL Sect. 13.1) and (ii) coding the meaning contribution of affixes as cat and sem values.

The function of the core value in the automatic word form recognition of a language other than English, here German wusste, may be shown as follows:

12.4.4 Owner-based storage of language proplet at now front

ii *memory component*

member proplets of A-*memory*	*now front*	*owners*	B-*memory*
. . .	$\begin{bmatrix} \text{sur:} \\ \text{verb: know} \\ \text{cat: n}' \text{ a}' \text{ v} \\ \text{sem: past ind} \\ \text{arg:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 37} \end{bmatrix}$	⇐ know	
		⇑	

i *interface component*

	recognized surface	*analyzed surface*	
raw input data	⇒ wuss-te ⇒	$\begin{bmatrix} \text{sur: wusste} \\ \text{verb: know} \\ \text{cat: n}' \text{ a}' \text{ v} \\ \text{sem: past ind} \\ \text{arg:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 37} \end{bmatrix}$	⇒ know

The automatic word form recognition of the hear mode begins with raw input data provided by a sensor of the interface component (i). A language-dependent word form type matching the raw data results in the recognized surface (⇒) wuss-te (letter sequence). It is used for lexical lookup of the analyzed surface, i.e. the complete proplet, from the allomorph trie structure (12.5.3). The core value know serves (a) to access (⇑) the token line of know and (b) to store (⇐) the proplet retrieved from the trie structure, but without the sur value, at the now front (ii). After processing, now front clearance leaves the proplet behind (loomlike clearance) in what is becoming permanent storage (member proplets).

12.5 String Search

The transition from the recognized surface to the analyzed surface in 12.4.4 requires lookup of a lexical entry in the agent's on-board memory. Computationally, the obvious solution for this task is string search. A precondition for string search is the medium of writing. In DBS, this medium is provided by the interface component which maps between raw data and the alphanumerical processing modality of a programming language (11.2.2).

The first step of implementing string search is assigning position numbers, here to the 26 letters[12] of English, resulting in a *numbered list*. The following example is from the Jane Austen novel EMMA, beginning of the 2nd paragraph.:

12.5.1 LETTERS OF A TEXT AS A NUMBERED LIST

```
             1         2         3         4
m: 01234567890123456789012345678901234567890...
S: She was the youngest of the two daughters...
```

Line S is the text to be searched. Its letters are numbered in line m, with each number directly above the associated letter. Turning an online text automatically into a numbered list is computationally straightforward and efficient.

Next, an *inverted file* specifies for each alphanumerical character all its positions in the numbered list, as in the following example:

12.5.2 INVERTED FILE FOR THE ENGLISH SENTENCE IN 12.5.1

a: 5, 33
b:

[12] For simplicity, the 10 digits of the number system are left aside.

c:
d: 32
e: 2, 10, 17, 26, 38
f:
g: 16, 26, 35
h: 1, 36
i:
j:
k:
l:
m:
n: 15
o: 13, 30
p:
q:
r: 39
s: 0, 6, 18, 40
t: 8, 19, 24, 28, 37
u: 14, 34
v:
w: 4, 29
x:
y: 12
z:

For example, t: 8, 19, 24, 28, 37 means that the 8th, 19th, 24th, 28th, and 37th letter in 12.5.1 is a t. If the user spontaneously decides to look for a word, e.g. two in the text of 12.5.1, typing a search command, e.g. CTRL s, and the first letter t of the word activates the inverted file line t: 8, 19, 24, 28, 37. This causes highlighting the corresponding occurrences of t in the online text on the screen. When the second letter w is typed, the inverted file line w: 4, 29 is activated and only positions of a t followed by a w are highlighted, here 28, 29. When the third letter o is typed, the inverted file line o: 13, 30 is activated and only positions of a tw followed by an o, here 28, 29, 30, are highlighted. Just as turning an online text automatically into a numbered list, turning a numbered list into an inverted file is computationally simple and efficient.

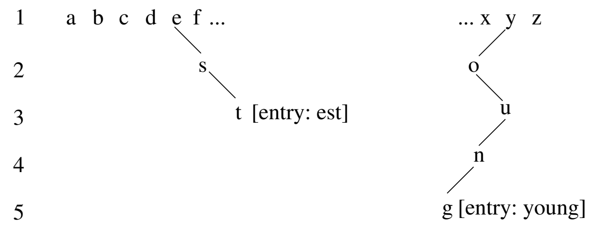
The method of string search was extended to building a *trie structure* as a lexical storage facility (Briandais 1959, Fredkin 1960, Flouri 2012).[13] For example, if a lexicon is stored in a trie structure, typing a search term (key) will bring up the associated lexical entry.

In the automatic word form recognition of DBS a trie is used for (a) segmenting a complex word form surface into its allomorphs and (b) looking up their definition (FoCL Sect. 14.3). The following example shows (i) the seg-

[13] From a linguistic point of view, tries are more transparent than hash tables because tries use the search terms' letter sequence directly, which supports the crucial segmentation into allomorphs.

mentation of youngest (12.5.1) into the allomorphs young and est,[14] and (ii) their linguistic definition as lexical entries (here shown in simplified form, i.e. [entry: young] and [entry: est]):

12.5.3 Example of a trie structure storing young and est



The top level shows the letters in their alphabetical order. Storage and retrieval of a lexical entry begins with the 1st letter of the input surface, here y. At the 2nd level, below the y, the 2nd letter of the input surface, here o, is connected to the 1st. At the 3rd level, the 3rd letter of the input surface, here u, is connected to the 2nd letter, here u, and so on. The traversal may[15] stop at a letter with a lexical entry, here the g at level 5.

As young- continues to young-est, the algorithm jumps back to the top level and walks down from the letter e to the final letter t of est to retrieve the next entry. The rules of computational morphology combine the analyzed allomorphs young- and -est automatically (data-driven) in time-linear order into the syntactic-semantic analysis of a complex word form (FoCL Sect. 14.4). The result is a lexical analysis like ‘superlative form of the adj young’ (categorization and lemmatization).

Trie search is general in that its only concern is the letter sequence of the search term, whereas the associated entry may be of any content, size, or form. Therefore tries may be used to access entries of any kind, astronomy, biology, chemistry, etc. At the same time, tries support the specific linguistic desideratum of automatically segmenting letter sequences into allomorphs (12.5.3) without any need for separators in the letter sequence (11.4.2, 11.4.3). This is the technical foundation of a rule-based morphological treatment of irregular paradigms and the on the fly analysis of neologisms (FoCL Chaps. 13–15).

In contradistinction to the statistical tagging popular in today’s corpus linguistics (FoCL Sect. 15.5), errors in a trie may be identified by automatically parsing test suites of arbitrary size. Once identified, errors and omissions may be corrected permanently. The trie method works for any writing system as

[14] Here the allomorphs equal the morphemes, i.e. the morphemes denote unit sets of allomorphs (FoCL 13.2.3).

[15] Alternative segmentations of a surface are possible (FoCL Sect. 14.3).

long as letters or strokes are in a total order, which seems to be the case in all writing systems.

12.6 Retrieval in the Speak Mode

The language counterpart of nonlanguage action[16] is language production in the speak mode. The following example shows the production of the German surface wusste from a *know* proplet stored in A-memory serving as input. As a content token it has a prn value, here 37, but no language-dependent sur value. The first step is shadowing the token to the now front (Sect. 6.5) as a proplet with the core value address (know 37) and the current prn value 84:

12.6.1 SHADOW-BASED PRODUCTION OF GERMAN SURFACE

ii *memory component*

member proplets of A-*memory*		*now front*	*owners*	B-*memory*
… [sur: verb: know cat: n′ a′ v sem: past ind arg: pro1 way mdr: nc: pc: prn: 37]	… ⇒	[sur: verb: (know 37) cat: n′ a′ v sem: past ind arg: pro1 way mdr: nc: pc: prn: 84]	know	

⇓

i *interface component*

		surface token		*type of surface form*
raw output data	⇐	wusste	⇐	verb: know –> wiss arg: pro1 way –> te sem: past ind –> wuss wuss-te

In this rough sketch, the shadowed proplet ((know 37) 84) is passed from the now front to the interface component (⇓). There, the speak mode performs the production of the correct language-dependent surface, here German wusste. Automatic word form production uses the features [verb: (know 37)], [arg: pro1 way], and [sem: past ind] of the *know* proplet to look up the corresponding German allomorphs wiss, wuss, and te for combination into the surface type wuss-te. Finally the surface type is adapted into a token and realized as the raw data of the unanalyzed external surface of the speak mode.

[16] The content provided by recognition must be stored, while blueprints for action must be retrieved. In this sense, there is an inherent connection between the recognition and storage of the resulting content (12.4.4), on the one hand, and the retrieval of a blueprint for action (12.6.1), on the other.b

13. Operation Component

The third main component of DBS cognition beside the (i) interface (Chap. 11) and the (ii) memory (Chap. 12) component is the (iii) operation component. It receives input from and writes output to the now front. Systematic now front clearance controls the data-driven application of operations by limiting input.

The operation kinds are (a) concatenation for recognition, (b) navigation for selective activation, and (c) inferencing for action. Concatenation builds content by connecting proplets with the semantic relations of structure coded by address, navigation travels along the semantic relations between the proplets in the agent's onboard database, and inferencing replaces input content with output content (Sect. 13.1).

The topic of Sect.13.2 is horizontal navigation along extrapropositional coordination, as when telling a story or reading a text. This is complemented in Sect. 13.3 by the vertical navigation along extrapropositional functor-argument. Sect. 13.4 presents shadowing as a means for stored content to participate in current processing at the now front. Sects. 13.5 and 13.6 summarize the functional flows of language and nonlanguage cognition.

13.1 The Operation Kinds of DBS

Operations are defined in the format of the declarative DBS specification. An operation is structured into (a) an antecedent, (b) a connective, and (c) a consequent. The operation kinds are alike in that their antecedent and consequent consist of pattern proplets. They differ in the number of input and output patterns, and in their connective. An operation application is triggered by matching input (data-driven) and derives an output.

The antecedent of a hear mode operation consists of two pattern proplets. The connectives are $\times$ called cross-copying, $\cup$ called absorption, and $\sim$ called suspension. The consequent of $\times$ and $\sim$n operations consists of two pattern proplets, while the consequent of $\cup$ operations consists of only one.

Consider the following example of a cross-copying operation:

13.1.1 HEAR MODE OPERATION SBJ×PRD

$$
\begin{array}{lcccccc}
\textit{pattern level} &
\left[\begin{array}{l}\text{sur:}\\ \text{noun: }\alpha\\ \text{cat: def sg}\\ \text{sem:}\\ \text{fnc:}\\ \text{prn: K}\end{array}\right] &
\left[\begin{array}{l}\text{sur:}\\ \text{verb: }\beta\\ \text{cat: n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg:}\\ \text{prn:}\end{array}\right] &
\Rightarrow &
\left[\begin{array}{l}\text{sur:}\\ \text{noun: }\alpha\\ \text{cat: def sg}\\ \text{sem:}\\ \text{fnc: }\beta\\ \text{prn: K}\end{array}\right] &
\left[\begin{array}{l}\text{sur:}\\ \text{verb: }\beta\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: }\alpha\\ \text{prn:K}\end{array}\right] \\
& \Updownarrow & \Uparrow & & \Downarrow & \Downarrow \\
\textit{content level} &
\left[\begin{array}{l}\text{sur:}\\ \text{noun: }\textbf{dog}\\ \text{cat: def sg}\\ \text{sem:}\\ \text{fnc:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 24}\end{array}\right] &
\left[\begin{array}{l}\text{sur:}\\ \text{verb: }\textbf{snore}\\ \text{cat: n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg:}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:}\end{array}\right] &
&
\left[\begin{array}{l}\text{sur:}\\ \text{noun: }\textbf{dog}\\ \text{cat: def sg}\\ \text{sem:}\\ \text{fnc: }\textbf{snore}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 24}\end{array}\right] &
\left[\begin{array}{l}\text{sur:}\\ \text{verb: }\textbf{snore}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: }\textbf{dog}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn:24}\end{array}\right]
\end{array}
$$

The second content proplet, here *snore*, is provided by automatic word form recognition and called the 'trigger proplet.' By matching (⇑) the second input pattern, called the 'trigger pattern,' the operation is activated to look at the now front for a content proplet matching its first input pattern (⇕). By binding α to dog and β to snore, the consequent produces (⇓) content proplets with the additional continuation features [fnc: snore] and [arg: dog], thus establishing the subject/predicate relation. This is computationally highly effective.

The second operation kind is navigation. It has one input pattern and one output pattern, and activates content in memory by traveling along the semantic relations of order-free proplets connected by address. By matching a content proplet, the input pattern computes a goal proplet as output. The navigation operations use eight connectives, namely ↙ and ↗ for subject/predicate, ↘ and ↖ for object\predicate, ↓ and ↑ for modifier|modified, and → and ← for conjunct−conjunct. Consider the following example:

13.1.2 NAVIGATING WITH V↙N FROM *snore* TO *dog* (arc 1)

$$
\begin{array}{lccc}
& \textbf{V↙N} & & \\
\begin{array}{l}\textit{pattern}\\ \textit{level}\end{array} &
\left[\begin{array}{l}\text{verb: }\alpha\\ \text{arg: }\beta\text{ X}\\ \text{prn: K}\end{array}\right] &
\Rightarrow &
\left[\begin{array}{l}\text{sur: lexnoun}(\hat{\beta})\\ \text{noun: }\beta\\ \text{fnc: }\alpha\\ \text{prn: K}\end{array}\right] \\
& \Uparrow & & \Downarrow \\
\begin{array}{l}\textit{content}\\ \textit{level}\end{array} &
\left[\begin{array}{l}\text{sur:}\\ \text{verb: snore}\\ \text{cat: \#n}'\text{ decl}\\ \text{b sem: past ind}\\ \text{arg: dog}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{array}\right] &
&
\left[\begin{array}{l}\text{sur: The dog}\\ \text{noun: dog}\\ \text{cat: snp}\\ \text{sem: def m}\\ \text{fnc: snore}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{array}\right]
\end{array}
$$

The example is shown in the speak mode because the language-dependent lexicalization rule in the sur slot of the output pattern is switched on.

The third operation kind is inference. Inferences have an open number of input and output patterns and are special in that they may apply forward (deductive, connective $\Rightarrow$) in the speak mode and backward (abductive, connective $\dashrightarrow$) in the hear mode. The following inference is called IMP-INT (7.3.1):

13.1.3 IMP-INT INFERENCE FOR STAR-0 STAR-1 MOOD ADAPTATION

$$\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: n-s3}'\text{ \#a}'\text{ impv}\\ \text{arg: pro2 }\beta\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{noun: }\beta\\ \text{cat: def sg}\\ \text{fnc: }\alpha\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K}\end{bmatrix}\Rightarrow\begin{bmatrix}\text{verb: }\alpha\\ \text{cat: \#n}'\text{ \#a}'\text{ interrog}\\ \text{sem: sbj }\textit{can}\\ \text{arg: pro2 (}\beta\text{ K)}\\ \text{prn: K+1}\end{bmatrix}\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: }\alpha\\ \text{prn: K+1}\end{bmatrix}\begin{bmatrix}\text{noun: }\beta\\ \text{cat: def sg}\\ \text{fnc: }\alpha\\ \text{prn: K+1}\end{bmatrix}\begin{bmatrix}\text{S: L}\\ \text{T: D}\\ \text{A: N}\\ \text{R: P}\\ \text{3rd:}\\ \text{prn: K+1}\end{bmatrix}$$

If $\beta\ \varepsilon$ {salt, pepper, potatoes}, then α = pass. If $\beta\ \varepsilon$ {window, door, hatch}, then $\alpha\ \varepsilon$ {open, close}, etc.

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

$$\begin{bmatrix}\text{verb: pass}\\ \text{cat: n-s3}'\text{ \#a}'\text{ impv}\\ \text{arg: pro2 salt}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{noun: salt}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{verb: pass}\\ \text{cat: \#n}'\text{ \#a}'\text{ interrog}\\ \text{sem: sbj }\textit{can}\\ \text{arg: pro2 (salt 52)}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{noun: pro2}\\ \text{cat: sp2}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{noun: salt}\\ \text{cat: def sg}\\ \text{fnc: pass}\\ \text{prn: 52}\end{bmatrix}\begin{bmatrix}\text{S: veranda}\\ \text{T: ...15:20}\\ \text{A: sylvester}\\ \text{R: speedy}\\ \text{3rd:}\\ \text{prn: 52}\end{bmatrix}$$

The STAR-0 of the input and the STAR-1 of the output content show the speaker speedy and the hearer sylvester on the veranda with sylvester saying to speedy Could you pass the salt?

The differences between hear mode, navigation, and inference operations are completely accommodated within the declarative specification based on pattern and content proplets, and the antecedent, connective, consequent format. As such, they have a straightforward translation into a general purpose programming language of choice.

13.2 Extrapropositional Coordination

In DBS, the data-driven computation of possible continuations raises two questions: (i) what triggers the start of a derivation sequence and (ii) what ends it? Consider the following example of a short text stored in memory:

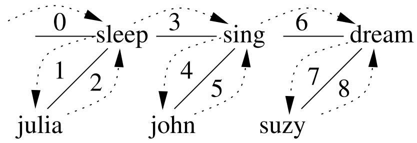
13.2.1 CONTENT OF Julia slept. John sang. Suzy dreamt.

$$\begin{bmatrix}\text{sur: julia}\\ \text{noun: [person x]}\\ \text{cat: snp}\\ \text{sem: nm f sg}\\ \text{fnc: sleep}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 11}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: sleep}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: [person x]}\\ \text{nc: }\textbf{(sing 12)}\\ \text{pc:}\\ \text{prn: 11}\end{bmatrix}\begin{bmatrix}\text{sur: john}\\ \text{noun: [person y]}\\ \text{cat: snp}\\ \text{sem: nm m sg}\\ \text{fnc: sing}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: }\textbf{sing}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: [person y]}\\ \text{nc: }\textbf{(dream 13)}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{sur: suzy}\\ \text{noun: [person z]}\\ \text{cat: snp}\\ \text{sem: nm f sg}\\ \text{fnc: dream}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 13}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: }\textbf{dream}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: [person z]}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 13}\end{bmatrix}$$

The content consists of a coordination of three maximally simple propositions: each conjunct consists of a name as subject and an intransitive verb in the past indicative ([sem: past ind]) as predicate. The coordination is coded by the extrapropositional addresses (sing 12) and (dream 13) serving as the nc values of *sleep* and *sing*, respectively.

Based on extrapropositional coordination, a single trigger may activate an autonomous navigation which continues indefinitely through connected contents in the agent's A-memory. The coordination stops (i) when no continuation value is provided by the nc slot of the current output proplet or (ii) assumed traversal counters in the proplets stored in the database signal redundant repetition. The traversal of 13.2.1 may be shown grapically as follows:

13.2.2 TRAVERSING AN EXTRAPROPOSITIONAL COORDINATION



In contradistinction to the bidirectional relations between conjuncts in intrapropositional coordination (3.3.2, arcs 5 and 6), extrapropositional coordination is unidirectional in that the nc, but not the pc, slots of the top verbs have values. Traversing connected propositions in the anti-temporal direction is possible, but requires the following inference:

13.2.3 INFERENCE NAVIGATING BACKWARD THROUGH A COORDINATION

$$\begin{bmatrix}\text{verb: }\beta\\ \text{pc: }\alpha\\ \text{prn: n+1}\end{bmatrix} \Rightarrow \begin{bmatrix}\text{verb: }\alpha\\ \text{nc: }\beta\\ \text{prn: n}\end{bmatrix}$$

The intrapropositional functor-arguments subject/predicate, object\predicate, and modifier|modified, as well as the coordinations noun−noun, verb−verb, and adj−adj are implemented from the outset for bidirectional traversal because it is always required. An extrapropositional V−V backward navigation, in contrast, is implemented by inference and used only when needed, as when telling a story starting from the end.

13.3 Extrapropositional Functor-Argument

The other extrapropositional construction of natural language besides coordination is functor-argument. Consider the following example:[1]

[1] An extrapropositional coordination counterpart is Lucy found a big blue square. She was happy.

13.3.1 Adverbial modifier construction as a set of proplets

Lucy was happy when she found a big blue square.

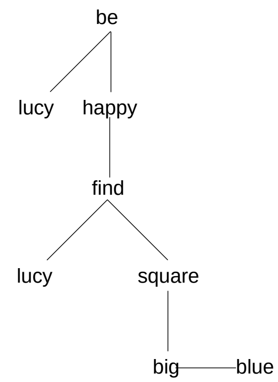
sur: lucy	sur:	sur:	sur:
noun: [person x]	verb: be	adj: happy	verb: find
cat: snp	cat: #ns13′ #be′ decl	cat: adn	cat: #n′ #a′ v
sem: nm f	sem: past ind	sem: pad	sem: *when* past ind
fnc:	arg: [person x]	mdd: be	fnc: be
mdr:	mdr: happy	mdr: (find 28)	arg: [person x] square
nc:	nc:	nc:	mdd: (happy 27)
pc:	pc:	pc:	nc:
prn: 27	prn: 27	prn: 27	pc:
			prn: 28

sur: lucy	sur:	sur:	sur:
noun: [person x]	noun: square	adj: big	adj: blue
cat: snp	cat: snp	cat: adn	cat: adnv
sem: nm f	sem: indef sg	sem: pad	sem: pad
fnc:	fnc: find	mdd: square	mdd:
mdr:	mdr:big	mdr:	mdr:
nc:	nc:	nc: blue	nc:
pc:	pc:	pc:	pc: big
prn: 28	prn:28	prn:28	prn:28

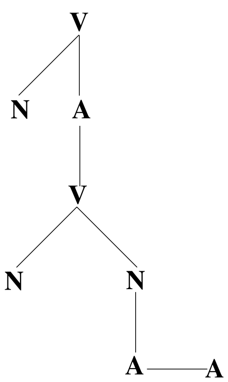
The semantic structure may be shown graphically as follows:

13.3.2 Canonical graph structure of 13.3.1

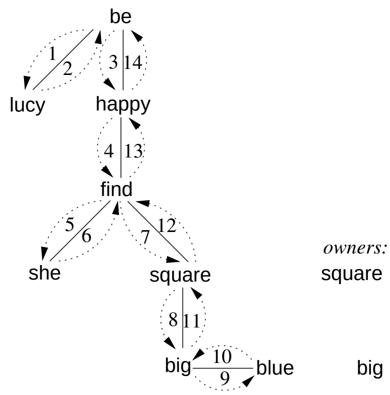
(i) SRG (semantic relations graph)



(ii) signature



(iii) NAG (numbered arcs graph)



(iv) surface realization

1	lucy	V/N
2	was	N/V
3	happy	V|A
4	when	A|V
5	she	V/N
6	found	N/V
7	a	V\N
8	big	N|A
9	blue	A_A
10		A_A
11	square	A|N
12		N\V
13		V|A
14	.	A|V

(v) activation sequence

7 square 8 big 9 blue 10 11 square 12 13 14
1 lucy 2 was 3 happy 4 when 5 she found 13 14

There are two ways of traversing a NAG, shown in *(iv)* and *(v)*. The (iv) surface realization is triggered by the narrative speak mode or by inferencing and starts from the top verb to produce a language-dependent surface.[2](e.g. 2.6.1, 7.3.3, 10.2.2). The coactivation of precedents (v), in contrast, is triggered by current content at the agent's now front, here square and big, and may resonate the entire content by continuing with selective activation.

If only one concept is used to resonate contents in a sizeable memory, the number of resonating contents will be (i) very large and (ii) highly divers. The method for reducing the yield and at the same time increasing the similarity with current now front content is intersection (Sect. 3.2). For example, for activating all contents matching the trigger concept be happy modified by find, the following intersection patterns is used:

13.3.3 Intersection patterns for finding be happy contents

$$\begin{bmatrix}\text{verb: be}\\ \text{arg: }\alpha\\ \text{mdr: happy}\\ \text{prn: K}\end{bmatrix}$$
$$\begin{bmatrix}\text{adj: happy}\\ \text{mdd: be}\\ \text{mdr: (find K+1)}\\ \text{prn: K}\end{bmatrix}$$

The patterns are derived from proplets in the manner described in 3.2.1 and moved along the token lines of be and happy similar to 3.2.2. Using higher and higher degrees of intersection (3.2.3) will increase conciseness until the only resonating contents equal the now front content completely.

In summary, for reasoning without surface production, the route through a content may start from any entrance point. For producing the language-dependent word order, surface forms, function words, and agreement, in contrast, the navigation must first go to the top verb to start production from there. In either case, the choice between continuation alternatives, if present, is guided by such factors as marked vs. unmarked, fore- vs. backgrounding, and theme-rheme (CLaTR Sect. 9.5).[3]

[2] In DBS reasoning, the production of language-dependent surfaces are optional, i.e. reasoning may proceed independently of surface production. In natural agents, this seems to amount to a measurable difference in processing load. For example, Fedorenko and Varley (2016) found (i) that patients who used to be able to speak but were later struck with aphasia, continued to perform arithmetic and similar tasks unimpaired and (ii) that non-aphasic subjects use the language areas in the brain only when processing language. In DBS terms, these would be the areas for language-dependent surface recognition and production (Sects. 11.4, 11.5); they require complicated linguistic analysis as well as substantial computational processing.

13.4 Data-Driven Adaptive Cognition

Just as the hammer mills, flouwer mills, saw mills, weaving mills, spinning mills, and grinding mills of the pre-industrial age were water-driven in that they used mid-sized streams as their source of energy, computational DBS cognition is data-driven: the incoming surfaces of the hear mode are data which drive the lexical lookup of proplets and their time-linear composition into content, the autonomous navigation of the speak mode uses the output of operation n as data which serve as input to the operation n+1, and similarly for the application of inferences.

The switching between these modes is data-driven as well. The hear mode is activated by incoming surfaces, the speak mode is activated by the hear and the think mode, and the think mode is driven by language and nonlanguage recognition, including the monitoring of the agent's own actions. Treating the repetition in these procedures uniformely as iteration provides a system-wide pulse, which supports not only smooth transitions between modes, but also the synchronization of communication between agents.

In addition to accommodating the different routines in a stable ecological niche, cognition must handle short term surprises and longterm changes. DBS deals with unusual change by coactivating precedents in memory which trigger matching inferences in the operation component. If this is not sufficient, conjunctions of relevant content may be used to automatically derive new inferences (Sect. 14.3).

A new inference is added to the operation component, while the content from which it was derived remains at the now front until clearance and then ends up as content in permanent storage. In memory, precedents triggering an inference, old or new, may be followed (i) by content matching the output pattern of inferences (3.6.5, 3.6.6) or (ii) by 'free' content. Either may be relevant for reasoning and learning. Because precedents in memory must be shadowed (6.6.2) to the now front in order to participate in current processing, data-driven operation applications are tightly controlled.

[3] In story telling, mapping a sequence of STAR-0 contents into a STAR-1 content requires the ability to integrate temporal and spatial STAR-0 specifications, which are lost in the STAR-0 STAR-1 transition (7.1.3–7.1.6), into the content (7.1.7, 7.1.8). Thereby the past STAR-0 contents' order may be reshuffled in the resulting STAR-1 contents using such operators as before that or in the mean time for foregrounding/backgrounding. The computational-linguistic reconstruction of these mechanisms is a rich empirical field beyond current scope.

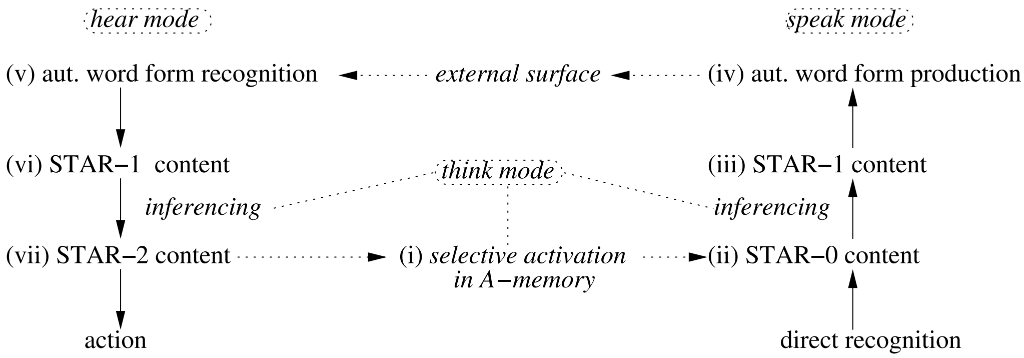
13.5 Functional Flow of Language Communication

There is communication with and without the use of a natural language. In natural language communication, the agents use external word form surfaces with cognition-internal literal meanings, attached by conventions which every member of the language community had to learn. In communication without a natural language, the agents may use endocrinic messaging based on hormones, exocrinic messaging based on pheromones, for example, in ants,[4] or samples, for example, in bees communicating a source of pollen.

The cognition of an agent with language may be used in recognition and action with and without language. For language communication, two agents (3.1.1, 3.1.2) are required, a speaker and a hearer, while activities without language may be confined to a single agent (3.1.3). Also, the hear and the speak mode require automatic word form recognition and production,[5] whereas the think mode may run without any language surfaces.

The DBS reconstruction of natural language communication resulted in the following seven trigger points:

13.5.1 TRIGGER POINTS IN THE LANGUAGE COMMUNICATION CYCLE



A (ii) STAR-0 content is produced from direct recognition or content shadowed from (i) A-memory to the now front.[6] The STAR-0 content may trigger (iii) a STAR-1 speak mode adaptation to the speaker's current communicative purpose (Sects. 7.3–7.5, Chap. 9). The (iv) automatic word form production

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

[5] If done right, word form recognition and production are not only quite demanding linguistically, but also require sizeable software components computationally. In natural agents, this is reflected by the large brain areas activated in language use.

[6] The think mode operations of selective activation and reasoning are triggered by content matching a search pattern (13.3.3), resulting in a selective activation without (13.3.2) or with the derivation of new content (5.6.2 ff.).

of language-dependent surfaces (11.5.2) is performed by lexicalization rules which sit in the sur slot of language proplets and are triggered when their proplet is activated during the think mode processing underlying the speak mode (2.6.2, 2.6.3).

A surface produced by the speaker triggers a lexical lookup (v) in the hearer by matching the raw data produced by the speaker, segmenting them into a sequence of allomorphs used for lexical lookup (12.5.3) and concatenation (11.4.3). As soon as a next word proplet is provided in this way, it is attached to or absorbed into the current sentence start (vi), i.e. what has been derived so far (2.1.3, 7.3.4, time-linear derivation order). The content resulting from the hearer's interpretation of the speaker's STAR-1 surface is adapted to the hearer's perspective (vii) by adjusting the interpretation of indexicals (8.2.1) and reverting nonliteral uses to the speaker's original impulse, if applicable (e.g. 8.3.2).

13.6 Functional Flow of Nonlanguage Cognition

There are two kinds of nonlanguage content: (i) language content which happens to lack the optional language-dependent sur values in the selective activation and inferencing of the think mode, and (ii) genuine nonlanguage content in (a) agents without language (but nevertheless able to communicate, e.g. for reproduction) and (b) when agents with or without language quietly observe. So far, we have only dealt with nonlanguage content (i).

From the viewpoint of evolution, the language ability grew out of genuine nonlanguage cognition. For building a talking robot, this constitutes the task of reconstructing genuine nonlanguage cognition as a electronic-mechanical system based on successful upscaling cycles in laboratories for artificial intelligence and robotics. However, recognition of and action in raw data data can not be done in isolation, but comes with the requirement of a clearly designed cognitive software structure (theory) which the machine input analysis (recognition) is mapped into and the raw data output (action) is mapped out of.

Without language-dependent function words, morphological variation, agreement, and word order, genuine nonlanguage content may be built (i) from the comparatively language-independent core and continuation concepts of language proplets (ii) concatenated semantically by simplified variants of the classical semantic relations of structure, i.e. subject/predicate, object\predicate, modifier|modified, and conjunct−conjunct. As an example consider how isolated cues in a two-and-half dimensional visual space (Marr 1982) might be composed into content.

Instead of *binding* variables to corresponding values in content proplets (hear mode), nonlanguage concatenation *substitutes* the variables[7] with recognition values in (copies of) nonlanguage recognition operations for visual cues. More specifically, when a concept value provided by the interface component (e.g. 1.3.2, 1.3.5) activates a genuine nonlanguage recognition operation by matching its second input pattern, the first pattern actively searches for a matching concept, not at the now front but in the agent's current visual space.[8] If one is found, the operation applies. For example, the recognition of a square matches the second pattern of the operation A|N (adjective modifying noun, 2.5.1):

13.6.1 Nonlanguage A|N operation in agent with language

$$
\begin{array}{lccccc}
\textit{operation pattern} &
\begin{bmatrix}\text{adj: a\_x}\\ \text{cat:}\\ \text{sem:}\\ \text{mdd: n\_y}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{noun: n\_y}\\ \text{cat:}\\ \text{sem:}\\ \text{fnc: v\_z}\\ \text{mdr: a\_x}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{adj: blue}\\ \text{cat:}\\ \text{sem:}\\ \text{mdd: square}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 47}\end{bmatrix} &
\begin{bmatrix}\text{noun: square}\\ \text{cat: sg}\\ \text{sem:}\\ \text{fnc: v\_z}\\ \text{mdr: blue}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 47}\end{bmatrix} \\
 & \Uparrow & \Uparrow & & \Downarrow & \\
\textit{input output} & \text{blue} & \text{square} & & \text{blue|square} &
\end{array}
$$

In the input, the substitution variables a_x and n_y are replaced by the concept tokens blue and square, resulting in a genuine nonlanguage modifier|modified relation blue|square by cross-copying in the output.[9] The use of full-fledged proplet patterns at the operation level makes the output suitable for further processing in an agent with language. The absence of a sur attribute indicates the nonlanguage origin of the content.

In agents without a natural language, e.g. dogs, a simpler version of the proplets may be assumed:

13.6.2 Nonlanguage A|N operation in agent without language

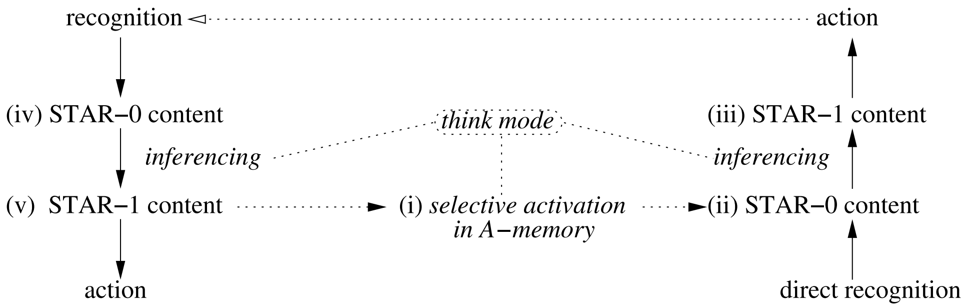
$$
\begin{array}{lccccc}
\textit{operation pattern} &
\begin{bmatrix}\text{adj: a\_x}\\ \text{mdd: n\_y}\\ \text{prn: K}\end{bmatrix} &
\begin{bmatrix}\text{noun: n\_y}\\ \text{fnc: v\_z}\\ \text{mdr: a\_x}\\ \text{prn: K}\end{bmatrix} &
\Rightarrow &
\begin{bmatrix}\text{adj: blue}\\ \text{mdd: square}\\ \text{prn: 47}\end{bmatrix} &
\begin{bmatrix}\text{noun: square}\\ \text{fnc: v\_z}\\ \text{mdr: blue}\\ \text{prn: 47}\end{bmatrix} \\
 & \Uparrow & \Uparrow & & \Downarrow & \\
\textit{input output} & \text{blue} & \text{square} & & \text{blue|square} &
\end{array}
$$

The step from genuine nonlanguage recognition in an agent without language 13.6.2 to genuine nonlanguage recognition in an agent with language 13.6.1

requires no more than adding the attributes of a corresponding full-fledged proplet.

In nonlanguage cognition, recognition and action take place in the same agent (leaving nonlanguage communication aside). Nevertheless, the cycle of nonlanguage cognition from recognition to action may be shown in analogy to the cycle of language cognition 13.5.1:

13.6.3 TRIGGER POINTS OF NONLANGUAGE COGNITION



The selective activation in (i) the think mode in language and in genuine nonlanguage cognition are alike in that they result in (ii) a STAR-0 content and are each complemented by direct recognition. The inferencing from (iii) a STAR-0 content to a STAR-1 content in language and in genuine nonlanguage cognition are also alike in that they adjust the agent's perspective on stored content and adapt current content to the agent's current purpose: in language communication, examples of the latter are softening a command into a polite request or strengthening a question into a command (Sects. 7.3–7.5); in genuine nonlanguage cognition, adaptation to a current purpose may be illustrated by the extra care when handling a precious fragile object.

The mapping from (iv) to (v) includes processing the outcome of the agent's own action (feedback, Sect. 14.5) and completes the cycle of nonlanguage cognition.[10] An example of the nonlanguage cognition cycle is the act of *looking closer* (11.6.2): it starts with the recognition of something seen before, continues with searching for related contents (e.g. experiences or something learned) from memory (coactivation, Sect. 3.1), and then rechecks with more

---

[7] Substitution variables originated in the hearmode, e.g. 2.1.2 (lines 1-5), but are just as suitable for substitution in nonlanguage patterns (NLC Sects. 6.1, 6.3).

[8] This may be supported by memory-based pattern completion, Hausser (2005).

[9] The substitution variables indicate obligatory values, while empty slots are used for optional values. Though modification is optional, there must be mdr and mdd values in order for the functor-argument operation 13.6.1 to apply.

[10] Without language, there is no STAR-1 STAR-2 adaptation of the hear mode (Chap. 8), which simplifies matters.

recognition. In DBS, this is treated systematically by (a) sensors controlling actuators and (b) actuators controlling sensors (Sect. 11.6).

14. Degrees of Abstraction

Conceptually, pattern matching in DBS is based on the type-token relation from philosophy. Computationally, tokens are defined as the data structure of proplets and generalized into types by replacing constants with variables and indexicals. Empirically, a pattern's degree of abstraction is controlled by the choice of variables and indexicals, and restrictions on their interpretation.

Sects. 14.1 and 14.2 demonstrate how variable restrictions may be used to automatically turn any content into a strictly equivalent pattern and any pattern into matching contents. Sect. 14.3 shows the on the fly derivation of a new inference. Sect. 14.4 integrates routine and exceptional behavior in the agent's ecological niche. Sect. 14.5 uses classic control theory for routine behavior. Sect. 14.6 explains polysemy and homonymy in terms of language change.

14.1 Computational Similarity

The limiting case of similarity is equivalence between a pattern and a content. It is based on the familiar method of defining variable restrictions (e.g. 5.5.2):

14.1.1 STRICT EQUIVALENCE BETWEEN A CONTENT AND A PATTERN

$$
\textit{pattern}\;
\begin{bmatrix}\text{sur:}\\ \text{noun: }\alpha\\ \text{cat: snp}\\ \text{sem: pl exh}\\ \text{fnc: }\beta\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{verb: }\beta\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: }\alpha\\ \text{mdr:}\\ \text{nc: (}\delta\text{ K+1)}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{noun: }\gamma\\ \text{cat: snp}\\ \text{sem: nm sg}\\ \text{fnc: }\delta\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K+1}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{verb: }\delta\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: }\gamma\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K+1}\end{bmatrix}
\quad
\begin{array}{l}\textit{variable restrictions:}\\ \text{where }\alpha\ \varepsilon\ \{\text{child}\},\\ \beta\ \varepsilon\ \{\text{sleep}\},\\ \gamma\ \varepsilon\ \{[\text{dog x}]\},\\ \delta\ \varepsilon\ \{\text{snore}\}\text{, and}\\ \text{K = 14.}\end{array}
$$

$$\Updownarrow$$

$$
\textit{content}\;
\begin{bmatrix}\text{sur:}\\ \text{noun: child}\\ \text{cat: snp}\\ \text{sem: pl exh}\\ \text{fnc: sleep}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{verb: sleep}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: child}\\ \text{mdr:}\\ \text{nc: (snore 15)}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}
\begin{bmatrix}\text{sur: fido}\\ \text{noun: [dog x]}\\ \text{cat: snp}\\ \text{sem: nm sg}\\ \text{fnc: snore}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 15}\end{bmatrix}
\begin{bmatrix}\text{sur:}\\ \text{verb: snore}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: [dog x]}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 15}\end{bmatrix}
$$

The content corresponds to Every child slept . Fido snored ., i.e. the coordination of two nonlanguage propositions, one with a content noun, the other with a name as subject. In the pattern, the core, continuation, and prn values are variables, but due to their restrictions, content and pattern are semantically strictly equivalent.

The content and the pattern proplets are well-formed because they use only classical semantic relations of structure (2.5.1, 2.5.2), here intrapropositional subject/predicate and extrapropositional conjunct−conjunct. The conversion from the content to the pattern substitutes all core, continuation, and prn values in the content simultaneously with variables (method one) and restricts each variable in the pattern to the single value which it replaced (method two).

The yield of a pattern relative to a memory may be systematically increased by adding values to the restriction sets of the variables. Consider a pattern matching the content corresponding to Every child slept, i.e. proposition 14 in 14.1.1, but with extended variable restrictions:

14.1.2 Converting a pattern into equivalent contents

$$\textit{pattern}\quad \begin{bmatrix}\text{noun: } \alpha\\ \text{cat: snp}\\ \text{sem: pl exh}\\ \text{fnc: } \beta\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} \begin{bmatrix}\text{verb: } \beta\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: } \alpha\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix} \quad \begin{array}{l}\textit{variable restrictions}\\ \text{where } \alpha\ \varepsilon\ \{\text{man, woman, child}\},\\ \beta\ \varepsilon\ \{\text{sleep, sing, dream}\},\\ \text{and K } \varepsilon\ \mathbb{N}\end{array}$$

$\Updownarrow$

	Every man slept.	Every man sang.	Every man dreamed.
contents	Every woman slept.	Every woman sang.	Every woman dreamed.
	Every child slept.	Every child sang.	Every child dreamed.

The nine contents are generated from the pattern by systematically binding the variables α and β to elements of their respective restriction sets.

Variable restrictions may be used (i) to extend the yield of a pattern derived from a single content (14.1.1) and (ii) to present partially overlapping contents as a single pattern:

14.1.3 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 (vertical) and the proplets *apple, pear, salad*, and *steak* occur four times as object (horizontal). Based on these repetitions, the sixteen propositions may be reduced automatically into a single equivalent pattern with the restricted variables α and β:[1]

14.1.4 REPRESENTING THE SET 14.1.3 AS A SINGLE 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 $\alpha\ \varepsilon$ {*Julia, John, Suzy, Bill*} and $\beta\ \varepsilon$ {*apple, pear, salad, steak*}

The Syntactic kinds (1.5.3), which are implicit in 14.1.3, are specified explicitly by the top attribute of each proplet in 14.1.4. The restriction sets of the variables α and β reduce the repetition of the four subjects and of the four objects in 14.1.3 to a single occurrence each.[2]

14.2 Controlling Degrees of Abstraction

The more concepts in a content are replaced with indexicals and variables, the higher the abstraction degree in the resulting pattern. Consider pattern proplets with increasing degrees of abstraction all matching the same content proplet:

14.2.1 REPLACING CONCEPTS WITH INDEXICALS OR VARIABLES

$$\begin{array}{l}\textit{pattern}\\ \textit{proplets}\end{array}\ 1\begin{bmatrix}\text{sur:}\\ \text{verb: sleep}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: child}\\ \text{mdr:}\\ \text{nc: (snore 15)}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\ 2\begin{bmatrix}\text{sur:}\\ \text{verb: sleep}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: child}\\ \text{mdr:}\\ \text{nc: X}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\ 3\begin{bmatrix}\text{sur:}\\ \text{verb: }\alpha\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: pro3}\\ \text{mdr:}\\ \text{nc: X}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\ 4\begin{bmatrix}\text{sur:}\\ \text{verb: }\alpha\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: }\beta\text{ W}\\ \text{mdr:}\\ \text{nc: X}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\ 5\begin{bmatrix}\text{sur:}\\ \text{verb: }\alpha\\ \text{cat: Y}\\ \text{sem: Z}\\ \text{arg: }\beta\text{ W}\\ \text{mdr:}\\ \text{nc: X}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}$$

matching

$$\begin{array}{l}\textit{content}\\ \textit{proplet}\end{array}\quad\begin{bmatrix}\text{sur:}\\ \text{verb: sleep}\\ \text{cat: \#n}'\text{ decl}\\ \text{sem: past ind}\\ \text{arg: child}\\ \text{mdr:}\\ \text{nc: (snore 15)}\\ \text{pc:}\\ \text{prn: 14}\end{bmatrix}$$

[1] The ε operator connecting a variable with its restriction set is used here in the interpretation ‘may be instantiated as.’

[2] The parsimony and high grammatical detail of 14.1.4 as compared to 14.1.3 results from the data structure of DBS. Linguistically, 14.1.4 may be seen as the valency pattern (Ágel et al. 2006) or lexical frame of the transitive verb *eat*. Restriction sets may be extended empirically by automatically searching a corpus with suitable patterns (CLaTR Sect. 15.5).

Though the five patterns are different, they match the same content proplet. The patterns' yield (CLaTR Sect. 6.5) relative to a suitably large memory will increase, however, with the abstraction as it increases from 1 to 5.

More specifically, pattern 1 is minimally abstract (maximally concrete) because the only variable is the prn value K; the extrapropositional feature [nc: (snore 15)] makes it unlikely that this pattern will yield more than the content from which is was derived. In pattern 2 the nc value is the variable X, making the pattern more abstract and consequently more likely to have a larger yield (depending on the size of the content in memory). In pattern 3, the core value sleep is replaced by the variable α and the continuation value child by the indexical pro3; depending on its variable restrictions and the 3rd value of the STAR, the yield may be substantial. In pattern 4, the continuation value child is replaced by the variable β, leaving only the grammatical attributes cat and sem with constant values. Pattern 5, finally, is maximally abstract (minimally concrete): the obligatory core, continuation and prn values are represented by the unrestricted variables α, β, and K, while the values in the other slots are represented by the variables W, X, Y, and Z.

Because of their different degrees of abstraction, the different patterns' overall yields (CLaTR Sect. 6.5) relative to a large A-memory will vary. The choice between concepts, indexicals, restricted variables, and unrestricted variables as core values directly affects search via the owners and their trie structure (12.5.3). If the core value of the pattern is a concept, the search is limited to a single token line. If it is an indexical, the search is limited to the token line of the associated STAR value. If it is a restricted variable, the search is limited to the token lines of the elements in the restriction set. If it is an unrestricted variable, however, the owners are of little use for limiting search.

For example, searching solely for a grammatical property, such as for all proplets with the core attribute verb or the cat value #sn3′, would require searching the complete A-memory. This is because (i) the proplets in a token line have different core attributes (CLaTR 6.6.4–6.6.7) and (ii) the arrival order of a proplet is unrelated to its grammatical values. Following standard practice in Computer Science, however, this kind of search may be made computationally efficient with little effort by automatically building indices for attributes or for grammatical values (relative to a given A-memory).

The converse of different patterns matching the same content is a single pattern matching different contents, provided the pattern is sufficiently abstract. The yield of an abstract pattern may be controlled precisely by replacing variables with concepts or by defining variable restrictions. In the following example, the only real restriction of the pattern is the core attribute verb. The input values are represented by the variables α, β, W, X, Y, Z, and K:

14.2.2 REPLACING VARIABLES WITH CONCEPTS

pattern proplet

$$\begin{bmatrix}\text{sur:}\\ \text{verb: } \alpha\\ \text{cat: Y}\\ \text{sem: Z}\\ \text{arg: } \beta \text{ W}\\ \text{mdr:}\\ \text{nc: X}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}$$

matching

content proplets

sur:	sur:	sur:	sur:	sur:
verb: sleep	verb: snore	verb: read	verb: give	verb: try
cat: #n′ decl	cat: #n′ decl	cat: #n′ #a′ decl	cat: #n′ #d′ #a′ decl	cat: #n′ #a′ decl
sem: past ind	sem: pres ind	sem: pres prog	sem: past ind	sem: past ind
arg: child	arg: dog	arg: child book	arg: man child apple	arg: child read
mdr:	mdr:	mdr:	mdr:	mdr:
nc: (snore 15)	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:
prn: 14	prn: 15	prn: 16	prn: 17	prn: 18

The pattern proplet equals pattern 5 in 14.2.1. As the most abstract, it matches all of the five content proplets shown.

14.3 On the Fly Derivation of a New Inference

The conversion between a content and a strictly equivalent pattern (14.1.1) and the control of abstraction levels in a pattern provide the technical means for the automatic derivation of new inferences from content. Like learning, acquisition of a new inference may require repeated experience, but may also be based on a single instance (Bandura 1986).

A constellation supporting the automatic formation of an inference is a content of the form A&B, where A and B are propositions and & indicates their ordered coordination. Constellations are similar if they use equivalent (CLaTR 12.6.3) semantic relations of structure (Sects. 2.5, 10.2) and corresponding concepts belong to the same semantic field (11.3.3).

Inducing an inference from an A&B content takes two steps. The first establishes similarity formally by deriving a pattern from the extrapropositional coordination. The second converts the coordination pattern A&B into the inference A⇒B by turning the feature [nc: (B K+1)] into the connective ⇒.

Consider the following derivation of the C-inference 3.5.1-3.5.2. The first step takes a currently activated content and turns it into a matching pattern by replacing the prn constants with variables:

14.3.1 STEP 1: CONVERTING A CONTENT INTO A PATTERN

$$\begin{array}{lcccc}
\textit{output pattern} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{nc: (bark K+1)}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{sur: fido}\\ \text{noun: [dog x]}\\ \text{fnc: bark}\\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: [dog x]}\\ \text{prn: K+1}\end{bmatrix}\\
 & \textit{conversion} \quad \Uparrow & & \Uparrow & \\
\textit{input content} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{nc: (bark 24)}\\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{sur: fido}\\ \text{noun: [dog x]}\\ \text{fnc: bark}\\ \text{prn: 24}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: [dog x]}\\ \text{prn: 24}\end{bmatrix}
\end{array}$$

The extrapropositional connection between the first and the second proposition pattern, expressed by the nc value (bark K+1) of *approach*, originates in the input content.

The second step derives the new inference from the pattern 14.3.1 by turning the feature [nc: (bark K+1)] into ⇒ and inverting the second ⇑ into ⇓:

14.3.2 STEP 2: INDUCTION TURNING PATTERN INTO DBS INFERENCE

$$\begin{array}{lccccc}
\textit{pattern level} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{nc: (bark K+1)}\\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{sur: fido}\\ \text{noun: [dog x]}\\ \text{fnc: bark}\\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: [dog x]}\\ \text{prn: K+1}\end{bmatrix}\\
 & \textit{induction} \quad \Uparrow & & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{noun: stranger}\\ \text{fnc: approach}\\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{verb: approach}\\ \text{arg: stranger}\\ \text{prn: 23}\end{bmatrix} & & \begin{bmatrix}\text{sur: fido}\\ \text{noun: [dog x]}\\ \text{fnc: bark}\\ \text{prn: 24}\end{bmatrix} & \begin{bmatrix}\text{verb: bark}\\ \text{arg: [dog x]}\\ \text{prn: 24}\end{bmatrix}
\end{array}$$

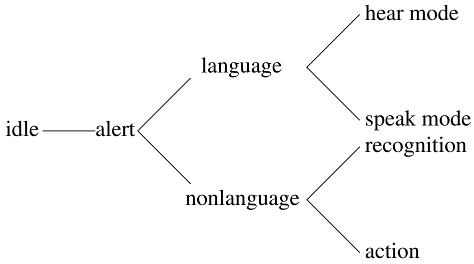
For induction in the traditional sense, one may require multiple instances of similar constellations as input to the automatic derivation of a new inference. This may be based on a counter: step 2 is only performed if the pattern derived in step 1 matches a certain number of similar contents in memory. A naive agent may be satisfied with two instances (jumping to conclusion), a more prudent one may require four or five.

The application of a DBS inference, old or new, is data-driven. Thus a newly derived inference requires no extra provision for integration into the system.

14.4 Autonomous Behavior Control

In a cognitive agent with language, the most basic options are between language and nonlanguage behavior, each followed by the options between recognition and action:

14.4.1 Most basic options of behavior control

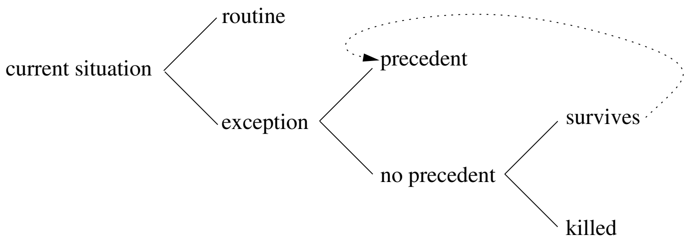


The functional flow begins with an initial state in which the cognitive agent with language is being idle (not preoccupied) but alert. Language recognition (hear mode) may be triggered by being talked to, which may subsequently trigger the speak mode; what is heard may determine what is being said (coactivation). Nonlanguage recognition (6.3.1), such as a feeling of hunger, may trigger an appropriate nonlanguage countermeasure as action (6.3.3).

As long as circumstances in the agent's ecological niche are 'normal,' the agent may maintain balance by following routines in B-memory (12.1.2). If the agent is challenged by an exceptional situation, however, balance may be lost. To regain it, a suitable reaction may have to be found on the fly and performed in real time. In DBS, this process involves reasoning based on inferencing with the support of pertinent contents in A-memory.

For complementing existing data-driven behavior control (Sects. 13.5, 13.6) with new behavior, the software is based on the following functional flow:

14.4.2 Functional flow for behavior without precedent



Once triggered, the software for a routine situation simply runs its course (CLaTR 14.4). An exceptional situation, in contrast, triggers a search for precedents in the agent's memory, using coactivation (Sect. 3.2) and selective activation (Sect. 3.3) with shadowing (Sect. 6.6) to the now front. Precedents resulting in success for the agent serve as positive models for similar situations in the future, while those resulting in failure serve as negative models.

If no precedent is found, the agent may react at random, as in fight or flight. If the agent's choice results in surviving the exceptional situation, there will be a precedent the next time (dotted arrow). Otherwise there will not be a next time for the agent and the experience is lost unless there were witnesses.

14.5 Using Control Theory for Maintaining Basic Balance

Like the elementary concepts in C-memory (11.3.2, 11.3.4), the complex concepts in B-memory (12.1.2) are rooted in the noncognitive domains of the natural sciences and the technologies of artificial recognition and action. B-contents are bare in that they are not assigned a STAR-0, but share the STAR of the A-content which they serve. Without monitoring the use of complex concepts, there is no need for B-memory to be writable (except for the occasional addition of another routine or a rare success in breaking a habit).

The implementation of B-contents may use aspects of control theory (Ogata 2010). The operations of traditional control theory are triggered by the moment-by-moment changes of a system's states, but without recording the changes. Without monitoring, control theory does not use STAR-0s, which is why it fits into the non-writable B-memory of DBS cognition.

Control theory proceeds from the distinction between open loop and closed loop systems, concentrating on closed loop systems. Even more basic is the distinction between systems with and without a loop. An example of a *no loop* system is a heater which has no mechanism for the automatic control of its activity. Instead, someone has to go down to the basement to turn the heater on or off; there is no system-internal memory.

An example of an *open loop* system is a furnace connected to a timer which may be programmed, for example, at low from 10 pm to 6 am and at mid from 6 am to 10 pm. Containing a clock and a small memory for storing the switch times, the timer assumes the role of raw data combined with a sensor.[3]

A *closed loop* system is based on feedback, which may be positive or negative. The purpose of negative feedback is to continuously maintain a certain target state. It is called negative because any deviation from the target value, higher or lower, is regarded as negative. Positive feedback, in contrast, provides for continuous expansion.

The distinctions between no_loop/loop, open-loop/closed-loop, and negative/positive feedback may be shown graphically as follows:

[3] The timer and the clocks in the house must be synchronized in order for the system to work as intended.

14.5.1 DISTINCTIONS OF CONTROL THEORY

no loop

open

control theory — loop

closed

negative feedback

positive feedback

A classic example of negative feedback is a thermostat (Wiener 1947). When the sensor measures that the current temperature exceeds the target value, the controller (thermostat) lowers the activity of the system (furnace) until the target value is reached; when the sensor value is below the target value, the controller raises the system activity until the target value is reached.[4]

Like negative feedback, positive feedback is a fixed behavior (CLaTR Sect. 6.1). In positive feedback, an item automatically divides at least into two, the results divide again, and again, and again, ... (exponential). An example is the number of births in a population of mice. Leaving other factors aside, an initial couple will produce six to eight babies in one litter which will soon produce their own offspring, which will result in another increase, and so on (population explosion).

Other examples of positive feedback are nuclear fission, global warming (Meadows 2008), and the collapse of a star. Without careful control, positive feedback leads to self-destruction as the opposite of balance. A practical application of positive feedback is the amplification of a weak radio signal to the desired level of strength, followed by negative feedback.

14.6 Polysemy and Homonymy

In language behavior, the transfer of content from a speaker to a hearer is achieved with nothing but modality-specific unanalyzed external surfaces (raw data) in their time-linear order (3.1.1, 3.1.2). Agent-externally these surfaces have no meaning and no grammatical properties at all, but they may be measured and analyzed by the methods of natural science. An agent competent in the language, however, recognizes the raw data as language-dependent surface forms which are attached by convention to lexical entries. These specify an appropriate core value and highly differentiated grammatical properties.

For adapting to the current communicative purpose, a speaker may choose a nonliteral use (Chaps. 7, 9). In order to understand the speaker correctly

[4] There are methods for smoothing between on and off states, such as "dithering" (Flügge-Lotz 1968).

(Chaps. 8, 9), the hearer must revert the nonliteral use into the literal version (in addition to adjusting indexicals). For this to work as a software mechanism, the literal meaning$_1$ of the elementary language signs and their syntactic-semantic composition must be clearly defined. While the literal meaning$_1$ of a given language sign is roughly the same for all members of the language community, the speaker meaning$_2$ differs in different utterances, from one hearer to another (Sects. 10.1), and from one interpretation to the next (Sect. 10.2).

Distinguishing different uses of the same literal meaning$_1$ in different utterance situations obviates misguided attributions of polysemy. Consider the following examples, which have been presented as "standard cases of polysemy" (Fogal 2016):

14.6.1 Alleged examples of polysemy

> line
> I drew a line; She read a line; He has lines around his eyes; Clothes hung on a line; Jorge waited in a line; I made a line of bad decisions.
>
> see
> Did you see the sunset?; I see your point; See how it sounds; You should see a doctor; See that you don't break it; Sam's been seeing Maxine.

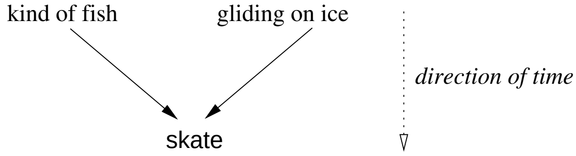
Instead of contaminating the meaning$_1$ of, e.g., line and see by postulating boundless lexical ambiguity with elements of use, the literal meaning$_1$ should be defined as minimal as possible and as precisely as necessary. All that is required of a meaning$_1$ is the ability *relative to a context of interpretation* to identify the intended referent in the case of nouns (6.4.1), specify the intended property in the case of adjs, prepnouns, and one-place verbs (6.4.2), and establish the intended relation in the case of two- or three-place verbs (6.4.3),

Spurious polysemy has the following disadvantages. First, computational inefficiency results in 14.6.1 from having to simultaneously pursue six different lexical readings for see and another six for line, resulting in six times six = 36 readings just for John saw a line alone. Second, each of the alleged meaning distinctions has to be provided with a declarative definition and an operational implementation (grounding, Sect. 1.3, 11.2.1), creating much more work than (ideally) only one literal meaning$_1$ per content word.

In classical lexicography the term polysemy is explained by contrasting it with the term homonymy. Synchronically a genuine polysemy and a genuine homonymy are alike in that each consists of a word form surface which has more than one literal meaning$_1$. Diachronically, however, polysemy and homonymy differ in how they evolve.

According to classical analysis, homonymy arises when different concepts happen to evolve the same surface:

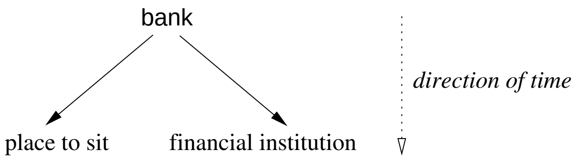
14.6.2 Evolution of a Homonymy



Other examples are stalk for (a) part of a plant and (b) harrassing a person, perch for (a) a kind of fish and (b) a place to roost, and left for (a) the past tense of leave and (b) the opposite of right.

Polysemy, in contrast, arises when the concept of a given surface, e.g. bank, evolves into different concepts in a process of specialization:

14.6.3 Evolution of a Polysemy



The origin of this polysemy is the practice of medieval money changers in Italy to display their wares on bench-like furniture. Other examples are book for (a) something to read and (b) making a reservation, wood for (a) part of a tree and (b) a forest, and crane for (a) a bird with a long neck and (b) equipment for lifting.

Homonymy and polysemy, on the one hand, and the shading of the speaker's utterance meaning$_2$, on the other, have in common that they depend on the context of interpretation (Sect. 6.1), i.e. the current utterance situation and/or resonating content in memory. They differ, however, in that the context of interpretation disambiguates a genuine homonymy or polysemy by *eliminating* one or more of their meaning$_1$ readings. The meaning$_2$ shadings of a sparse literal meaning$_1$, in contrast, are *contributed* by the context of interpretation.

15. The Fundamental Alternatives

This chapter summarizes the empirical, methodological, and computational aspects which separate (i) the data-driven agent-based ontology from (ii) the substitution-driven sign-based ontology. Sects. 15.1–15.5 consider whether or not the input and output of a cognitive agent should be processed in a strictly time-linear order; the associated dichotomies are computing (1) continuations instead of substitutions, treating repetition as (2) concatenation instead of embedding, and using (3) iteration instead of recursion. Sect. 15.6 concludes with the nature of memory: (a) why is a memory essential in an agent-based system but has no place in sign-based systems? (b) what follows from the distinction between a content-addressable and a coordinate-addressable memory? and (c) which is preferable?

15.1 Continuation instead of Substitution

On the one hand, the recognition and production of language, and of behavior in general, is time-linear. On the other hand, the syntactic-semantic structures of content are hierarchical. This ‘squaring of the circle’ in cognition is adumbrated in the following quotations from an early classic of modern linguistics:

> The configurations of behavior, however, tend to be predominantly temporal – it is the sequence of motions that flows onward so smoothly as the creature runs, swims, flies, talks, or whatever. What we must provide, therefore, is some way to map the cognitive representation into the appropriate pattern of activity. (p.13)
>
> It is quite obvious that the behavior is organized simultaneously at several levels of complexity. We shall speak of this fact as the “hierarchical organization of behavior.” (p.15)
>
> *Plans and the Structure of Behavior*
> Miller, Galanter, and Pribram (1960)

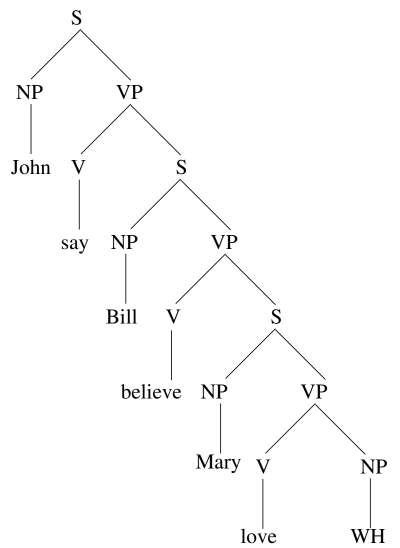
How can a static semantic hierarchy systematically produce a “sequence of motions”? As a first step towards an answer, agent-based DBS divides the question into two: (i) how can time-linear recognition build a static hierarchy and (ii) how can time-linear action be realized from such a static hierarchy?

To illustrate the switching between the static and the dynamic aspects of grammatical analysis in different formalisms, let us compare the PSG (Phrase Structure Grammar, Chomsky 1965) and the DBS analysis of the unbounded dependency construction corresponding to Whom does John say that Bill believes that Mary loves?

The construction is unbounded because the number of object sentences intervening between the top and the bottom clause has no grammatical limit. Thus ...John says that Bill believes that Mary loves WH may be extended into ...John says that Bill believes that Mary claims that Suzy loves WH, and so on. The dependency is the object\predicate relation between the surface-initial whom and the surface-final loves.

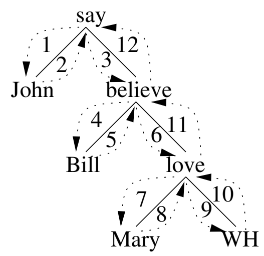
15.1.1 Comparing unbounded Dependency in PSG and DBS

PSG



DBS

numbered arcs graph (NAG)



surface realization

3	6	9	10	11	12	1	2
		Whom			does	John	say
V\V	V\V	V\N	N\V	V\V	V\V	V/N	N/V

3	4	5	6	7	8	11	12
that	Bill	believes	that	Mary	loves		?
V\V	V/N	N/V	V\V	V/N	N/V	V\V	V\V

The PSG analysis assumes that the WH at the bottom of the "deep structure" is transported by a sequence of movement transformations[1] to the initial position at the top ("fronting" of the WH object). This results in a "surface structure" in which John said that S is adjusted to Whom did John say that S.

DBS, in contrast, distinguishes between the speak and the hear mode. The speak mode leaves the numbered arcs graph derived in the hear mode (lab-

[1] Movement transformations were intended in part to reconcile constituent structure (Bloomfield 1933) with the "discontinuous elements" (Bar Hillel 1964, p. 102) found in natural language (FoCL Sect. 8.5). Unfortunately, complementing context-free PSG with a transformational component raised the computational complexity from polynomial to undecidable, which is not plausible for natural language (Harman 1963, Gazdar 1981). The undecidability of transformational grammar was formally proven by Peters and Richie, and published in 1973.

oratory set-up, Sect. 2.6) unchanged, and handles language production as a navigation along the semantic relations between nodes (*surface realization*).

Would it be possible to use the DBS navigation solution for PSG trees? One reason[2] why it works in DBS is the semantic interpretation of the graph lines, i.e. subject/predicate, object\predicate, adnomial|noun, and adverbial|verb (functor-argument), and conjunct−conjunct (coordination), at the elementary, phrasal, and clausal levels of grammatical complexity. The lines in a phrase structure tree, in contrast, represent *dominance* and *precedence*. These primitive notions are nonsemantic, undifferentiated, and completely unsuitable for supporting a semantically interpreted navigation.

15.2 Equivalent Speak Mode Input and Hear Mode Output

One standard of the DBS approach to computational cognition is functional equivalence at a certain level of abstraction between the natural and the artificial agent (Sect. 1.1). This allows for major differences between the two.

For example, in natural language cognition the primary sensory medium of the natural prototype is speech with the modalities of vocalization and audition (11.2.1). The primary sensory medium of a DBS robot, in contrast, is writing with the modalities of manipulation and vision as a precondition for using a programming language as the processing medium (11.2.2). Other examples of functional equivalence at a certain level of abstraction are the use of electrical motors instead of muscles for locomotion[3] and manipulation, and similarly for computer vision, computer audition, speech synthesis and recognition, and their natural counterparts.

To constrain functional equivalence, it is integrated into upscaling cycles for managing the complexity of building a computational cognition. Based on a test suite, a cycle expands the system to a concrete phenomenon, for example, subject-predicate agreement in English. Once functional equivalence is achieved in such a case, the next cycle is started with a more comprehensive test suite. In this way, data coverage is systematically verified and extended.

However, functional equivalence achieved by a current cycle does not guarantee that upscaling to the next cycle will be successful. For example, upscaling from NEWCAT (1986) to CoL (1989) seemed initially successful in that a substantially extended test suite ran free from error. Yet the attempt to next

[2] The other reason is the necessary distinction between recognition and action in agent-based DBS, which is not available in sign-based generative grammar (ontology-induced empirical omission).

[3] An example is Boston Dynamics' Atlas robot (150 kg) doing a backflip (Owano 2017, Simon 2017b).

upscale CoL was unsuccessful because its sign-based ontology (CoL Chap. 3) lacked the distinction between recognition and action.

To overcome this broken upscaling, DBS developed the heuristic principle of *input-output equivalence* (NLC 2nd ed., 1.5.2) between the natural prototype and an artificial model. The principle is founded on the fact that natural communication is successful if, and only if, the meaning$_1$ resulting from the interpretation by the hearer (input, recognition) equals the meaning$_1$ produced by the speaker (output, action) – which necessitates an agent-based approach:

15.2.1 Input-Output Equivalence in language communication

> Language processing by the natural prototype and an artificial counterpart are input-output equivalent if, and only if, they
> (i) take the same input and produce the same output,
> (ii) disassemble input and assemble output in the same way, and
> (iii) process the parts during recognition and action in the same order, i.e. from beginning to end.

The output of the speak mode and the input of hear mode is concretely manifested by a single external sequence of word form surfaces (raw data). This sequence provides the obvious order for hear and speak mode processing.

As the link between the speak and the hear mode in turn taking (Schegloff 2000), DBS uses the time-linear processing order also for the think mode. The alternative would be switching from a continuation-based hear mode to a substitution-based think mode, and back to a continuation-based speak mode.[4]

The change from the sign-based approach of SCG, NEWCAT, and CoL via FoCL to the agent-based approach of NLC, CLaTR, HBTR, and TExer provided not only an opportunity but also a need for reconstructing a host of traditional as well as new topics in a computationally viable form:

15.2.2 Design requirements for computational DBS cognition

1. Definition of the data structure of proplets (Sect. 1.4) and the database schema of the content-addressable A-memory (Sect. 1.5) in interaction with the time-linear algorithm (Sect. 2.2) of LA-grammar (LAG);

[4] Apart from horrendous inefficiency, this would prevent a basic design feature of DBS, i.e. the treatment of the speak mode as an optional mirroring of think mode navigation in the form of a sequence of language-dependent surfaces. A time-linear derivation order throughout DBS cognition is also supported by the fact that the C1-LAG subclass of the computational complexity hierarchy (TCS) parses the structures of natural language (FoCL 12.5.7), and by implication of thought, in linear time.

2. Definition of an agent-internal on-board orientation system (Sect. 11.6) for anchoring current recognition and action content to the current STAR-0 (Sect. 6.3) and providing values for indexicals to point at (Sect. 7.1);
3. Building (i) the interface (Chap. 11), (ii) the memory (Chap. 12), and (iii) the operations components (Chap. 13) of an artificial cognitive agent;
4. Reconstructing the natural sign kinds within the data structure of proplets and their computational interpretation as the Content kinds (a) concept implemented as matching, (b) indexical implemented as pointing, and (c) name implemented as baptism (Sects. 1.5, 6.4; CTGR);
5. Reconstructing the classical semantic relations of structure, i.e. functor-argument and coordination (Sect. 2.5), in terms of proplet-internal attributes and values defined as addresses;
6. Modeling the hear (Sect. 2.2), the think (Sect. 3.4), and the speak (Sect. 2.6) modes as procedures which are autonomous, but interact in the cycle of natural language communication (NLC Sect. 3.5);
7. Using the same data structure for language and nonlanguage content to treat generalized reference and coreference as purely cognitive processes (Sects. 3.1, 6.4, 6.5);
8. Defining similarity computationally as patterns with flexible degrees of abstraction (Sects. 14.1, 14.2), enabling content in memory to resonate with current processing (Sects. 3.2, 3.3);
9. Defining the inferences for syntactic mood adaptations in the speak (Chap. 7) and the hear (Chap. 8) mode;
10. Defining the inferences for the interpretation of figurative use in the speak and the hear mode (Chap. 9);
11. Computing the STAR-1 perspective of the speaker onto stored content (Chaps. 7, 9);
12. Computing the STAR-2 perspective of the hearer onto the interpretation of STAR-1 surfaces (Chaps. 8, 9).

The goal of the preceding chapters was to integrate this diversity of tasks into a coherent software design for a computational cognition with today's means.

15.3 Procedural instead of Metalanguage-Based Semantics

Before the advent of computers, the only scientific method of semantic interpretation were metalanguage definitions for the "object language," i.e. the language to be analyzed. This inherently sign-based method evolved over more than two millenia and was logically analyzed by Tarski (1935, 1944) in its definitive form: the metalanguage must be (i) formally constructed, (ii) its concepts must be mathematically obvious, and (iii) the object language may not contain the predicates true and false (FoCL Chaps. 19–21).

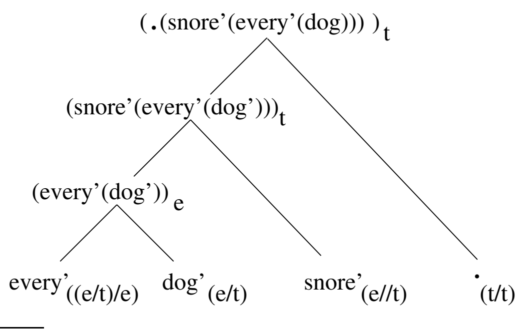
Conditions (i) and (ii) avoid circular paraphrasing.[5] Condition (iii) is based on Tarski's proof, founded on the 'strong' version of the Epimenides paradox designed by Leśniewski (1929), that the object language would otherwise be logically inconsistent. It follows that a truth-conditional analysis of natural language as the object language is necessarily incomplete,[6] according to its own foundation.

The advent of computers provided a basic alternative to Tarski's metalanguage method for semantic interpretation, namely the automatic execution of software procedures in a procedural semantics, based in DBS on the classical semantic relations of structure and the distinction between concepts, indexicals, and names. Unlike a truth-conditional semantics, a procedural semantics has no need for a principled exclusion of truth predicates, but requires computational viability; DBS also requires input-output equivalence (15.2.1).

Despite its known limitations, sign-based truth-conditional semantics continued as the favorite methodology in linguistics and philosophy, and culminated in Montague's quasi-automatic[7] translation of a "fragment of natural language" into a Predicate Calculus spruced up with (1) a typed λ (lambda) calculus (Church 1932), (2) intensions and extensions (Carnap 1947; FoCL Sect 20.2), and (3) a model structure with infinite sets of possible worlds and moments of time (C.I. Lewis and Cooper 1932) as the universe of discourse.

The Montague Grammar defined in PTQ uses two derivations for the syntactic-semantic analysis of a single input surface. The first is a bottom up Categorial Grammar derivation which analyzes the surface syntactically:

15.3.1 Categorial Grammar derivation of Every dog snores.



[5] Noted earlier by de Saussure (1916/1972), circular paraphrasing has been explicated further by D. Lewis (1970). In DBS, it is avoided by grounding elementary concepts in raw data (11.3.2, 11.3.4).

At first glance, the derivation appears to be time-linear. However, in concord with Polish Notation, the functors snore′ and • are added at the beginning of their respective sentence start, which violates time-linearity.

The second PTQ derivation begins with translating each word form of the syntactic derivation into a λ-expression (lexical lookup), followed by the top-down λ*-reduction* for semantic interpretation:

15.3.2 λ-REDUCTION FOR Every dog snores

$$(1)\ \widehat{p}[p]\ (\widehat{P}\ [P\widehat{x}(snore'(x))]\ (\widehat{Q}\widehat{P}\forall y[\underset{\uparrow}{Q}(y) \rightarrow P(y)]\ \underline{(dog')}))$$

$$(2)\ \widehat{p}[p]\ (\widehat{P}[\underset{\uparrow}{P}\widehat{x}(snore'(x))]\ (\underline{\widehat{P}\forall y[dog'(y) \rightarrow P(y)]}))$$

$$(3)\ \widehat{p}[p]\ \widehat{P}\ \forall y\ [dog'(y) \rightarrow \underset{\uparrow}{P}(y)]\underline{(\widehat{x}(snore'(x))}))$$

$$(4)\ \widehat{p}[p]\ \forall y\ [dog'(y) \rightarrow \widehat{x}(snore'(\underset{\uparrow}{x}))\underline{(y)}])$$

$$(5)\ \widehat{p}[\underset{\uparrow}{p}]\ \underline{\forall y\ [dog'(y) \rightarrow snore'(y)])}$$

result: $\forall y\ [dog'(y) \rightarrow snore'(y)]$

The lexical lookup in line 1 uses the bracketing structure and order resulting at the top of 15.3.1. The λ-translations in lines 2-5 are successively reduced into a formula of first order Predicate Calculus as the ready-made semantic interpretation. In each line, the underlined expression replaces the variable marked by $\uparrow$. For example, in line 2 $\underline{\widehat{P}\forall y[dog'(y) \rightarrow P(y)]}$ replaces P (SCG 29b).

Combining vertical bottom up and top down derivations for analyzing a single proposition of natural language is not input-output equivalent (15.2.1) with the natural prototype. Also, the meaning analysis of contingent concepts is

[6] As Tarski's student at UCLA, Montague must have been well aware of this. He preferred to be silent, however, perhaps because this fatal flaw arises for all formal systems of natural language which use a truth-conditional semantics. Montague's choice of the term *fragment* for his formal systems of natural language may be interpreted not only as referring to very limited data coverage, but also as a cunning hint at the systems' inherent incompleteness, i.e. the logically necessary absence of the words true and false in the object language. This is not a minor omission, because without the words true and false in natural language, a logical semantics could not be defined for it in the first place.

[7] Quasi-automatic as in a proof, i.e. strictly formal but unsuited for programming (Weyrauch 1980). For example, the attempt to program the canonical Montague Grammar defined in SCG revealed properties which look elegant intuitively, like 15.3.2, but are pointless from a software as well as a functional point of view (NEWCAT, Introduction). We know of no running program for a nontrivial fragment of Montague Grammar produced in the last 50 years.

unresolved: for example, dog′ and snore′ are just as arbitrarily defined as blue and square in 1.2.3. Moreover, the different semantic functions of elementary concepts such as dog′ (noun, referent), black′ (adj, property), and find′ (verb, relation) are flattened into mini-propositions with variables:

15.3.3 Noun, verb, and adj flattened into mini-propositions

the noun dog is interpreted as x is a dog and written as dog′(x)
the adj black is interpreted as x is black and written as black′(x)
the 1-place verb snore is interpreted as x snores and written as snore′(x)
the 2-place verb find is interpreted as x finds y and written as find′(x, y)
the 3-pl. verb give is interpreted as x gives y z and written as give′(x, y, z)

On the one hand, this allows to represent Every dog finds a bone as three mini-propositions which are coordinated with the propositional operators $\rightarrow$ and $\wedge$, and which have the variables bound by the quantifiers $\forall$ and $\exists$:

15.3.4 Every dog finds a bone in Predicate Calculus

$$\forall x[dog(x) \rightarrow \exists y[bone(y) \wedge find(x, y)]]$$

On the other hand, a uniform treatment of nouns, verbs, and adjs as minipropositions with variables is semantically (i) misguided because it loses the classical distinction between referents, properties, and relations (1.5.3). This (ii) obscures the empirical fact that relations and properties do not refer. Also, the use of coordination (based on the propositional operators $\rightarrow$ and $\wedge$) and coreference (based on the quantifiers $\forall$ and $\exists$) is (iii) contrived because there is neither coordination nor coreference in Every dog finds a bone.

Furthermore, if an existential and a universal quantifier occur in the same formula, they are treated as a source of ambiguity, depending on their order. For example, the following readings are attributed to Every man loves a woman:[8]

15.3.5 Readings attributed to Every man loves a woman

Reading 1: $\forall x\ [man(x) \rightarrow \exists y\ [woman(y) \wedge love(x,y)]]$
Reading 2: $\exists y\ [woman(y) \wedge \forall x\ [man(x) \rightarrow love(x,y)]]$

On reading 1, it holds for every man that there is some woman whom he loves. On reading 2 (which entails reading 1), there is a certain woman, e.g. Marilyn

[8] Montague (1970), English as a formal language (Sect. 6.7), uses a similar example. In Montague's terms, the verbs find in 15.3.4 and love in 15.3.5 are both "extensional" (FoCL Sect. 20.3).

Monroe, who is loved by every man. Attributing reading 2 also to 15.3.4 would result in There is a bone which every dog finds, which is not what Every dog finds a bone means.

The corresponding DBS analysis is (i) unambiguous and (ii) uses only intrapropositional functor-argument:

15.3.6 UNAMBIGUOUS ANALYSIS AS A DBS CONTENT

sur:	sur:	sur:
noun: man	verb: love	noun: woman
cat: snp	cat: #ns3 #a decl	cat: snp
sem: pl exh	sem: pres	sem: indef sg
fnc: love	arg: man woman	fnc: love
prn: 4	prn: 4	prn: 4

As in natural language, but unlike the formulas 15.3.4 and 15.3.5, there is neither coordination nor coreference. The determiner aspect of the quantifier $\forall$x representing every is coded as the features [sem: exh pl] (exhaustive plural) and [cat: snp] (singular noun phrase) of the *man* proplet, while the determiner aspect of the quantifier $\exists$y representing a(n) is coded in DBS as the feature [sem: indef sg] (indefinite singular) of the *woman* proplet. The semantic relations between the proplets are coded by intrapropositional address. The order-free proplets are held together by a shared prn value, here 4.

Another empirical inadequacy of Predicate Calculus are meaningful grammatical constructions which cannot be properly expressed. An example of such an empirical gap is Every farmer who has a donkey beats it, known as the *donkey sentence* (Geach 1962).[9] A systematic compositional derivation based on substitution results in the following representation in Predicate Calculus:

15.3.7 INCORRECT ANALYSIS OF WELL-FORMED SENTENCE

$$\forall x[[\text{farmer}(x) \wedge \exists y[\text{donkey}(y) \wedge \text{own}(x,y)]] \rightarrow \text{beat}(x,y)]$$

While the English sentence is well-formed and meaningful, 15.3.7 is inadequate because the y in beat(x,y) is not in the scope of $\exists$y binding donkey(y).[10]

The DBS alternative relies on treating all semantic relations of structure, including coreference (Sect. 6.5), by means of proplet-internal address values:

[9] The donkey sentence goes back to the middle ages (Walter Burley 1328/Gualterus Burlaeus 1988). It is only one of several instances in which the creaking hinges of Predicate Calculus' exoskeleton can not provide the correct quantifier scope. Discourse Representation Theory (Kamp 1980, Kamp and Reyle 1993) was developed in part as a solution to the donkey sentence.

[10] There have been numerous proposals to simply front the existential quantifier. In this way, donkey(y) would get bound by $\exists$y but at the cost of losing compositionality (King and Lewis 2017), which is methodologically unacceptable.

15.3.8 TREATMENT OF THE DONKEY SENTENCE IN DBS

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: ∅ 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

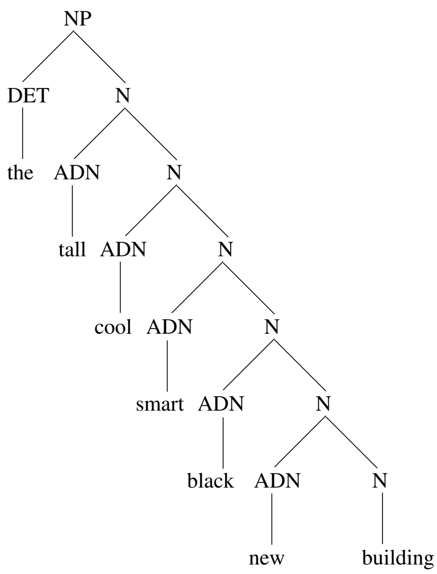
The embedding is reflected by the prn values, which are incremented from 16 to 17 in the transition from the main to the subordinate clause and decremented from 17 to 16 in the transition back to the main clause. The coreferential pronoun it in the main clause is represented by the extrapropositional address (donkey 17), which refers to the antecedent[11] in the subclause.

15.4 Concatenation instead of Embedding

Parallel to the alternative between computing possible continuations instead of possible substitutions is the alternative between repetition by concatenation instead of by embedding.[12] Repeating by embedding is used in PSG, and may be shown explicitly by the equivalent feature structure of HPSG (Pollard and Sag 1987, 1994). The following example uses repeating adnominals:

15.4.1 PSG TREE AND EMBEDDING IN EQUIVALENT HPSG ANALYSIS

PSG tree



HPSG feature structure

$$\left[\begin{array}{ll}\text{NP} & \left[\begin{array}{ll}\multicolumn{2}{l}{\text{DET: the}}\\ \text{N} & \left[\begin{array}{ll}\multicolumn{2}{l}{\text{ADN: tall}}\\ \text{N} & \left[\begin{array}{ll}\multicolumn{2}{l}{\text{ADN: cool}}\\ \text{N} & \left[\begin{array}{ll}\multicolumn{2}{l}{\text{ADN: smart}}\\ \text{N} & \left[\begin{array}{ll}\multicolumn{2}{l}{\text{ADN: black}}\\ \text{N} & \left[\begin{array}{l}\text{ADN: new}\\ \text{N: building}\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]$$

[11] In linguistics, the term 'antecedent' is used not only for inferences, but also for the full noun referent, here a donkey, preceding a coreferential pronoun, here it (CLaTR Chap. 11).

The repeating N[ADN N] substructure appears on the left as a vertical (diagonal) PSG tree, defined in terms of dominance and precedence, and on the right as the equivalent recursive feature structures of HPSG. Both representations are generated by the same PSG, defined as follows:

15.4.2 PSG FOR REPEATING ADNOMINALS

1. NP $\rightarrow$ DET N
2. N $\rightarrow$ ADN N
3. DET $\rightarrow$ the
4. ADN $\rightarrow$ tall cool smart black new
5. N $\rightarrow$ building

Rewrite rule 2 provides the repeating substitution (recursion).

Instead of computing possible substitutions, DBS treats the ADN repetition (NLC 15.2.3) as a flat coordination based on computing continuations:

15.4.3 COORDINATION OF ADNOMINALS AS A SET OF PROPLETS

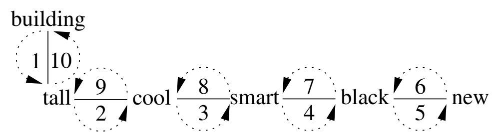
sur:	sur:	sur:	sur:	sur:	sur:
noun: building	adj: tall	adj: cool	adj: smart	adj: black	adj: new
cat: snp	cat: adn	cat: adn	cat: adn	cat: adn	cat: adn
sem: def sg	sem: pos	sem: pos	sem: pos	sem: pos	sem: pos
fnc:	mdd: building	mdd:	mdd:	mdd:	mdd:
mdr: tall	mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc: cool	nc: smart	nc: cool	nc: cool	nc:
pc:	pc:	pc: tall	pc: black	pc: new	pc: black
prn: 99	prn: 99	prn: 99	prn: 99	prn: 99	prn: 99

The modified *building* and the initial modifier *tall* are connected with the features [mdr: tall] and [mdd: building]. *Tall* starts the coordination with the feature [nc: cool]. As the beginning of the chain, the pc slot of *tall* is empty, indicating that there is no previous conjunct, just as the empty nc slot of the final conjunct *new* indicates that there is no next conjunct.

The semantic relations in 15.4.3 have the following graphical representation:

15.4.4 ADN REPETITION AS ANALYZED IN DBS

numbered arcs graph (NAG)



12 Whether or not repetition in the natural languages is based on embedding is the central issue in the debate between Hauser, Chomsky, and Fitch (2002) on the one hand and Everett (2005, 2013) on the other. For discussion see the following section 15.5.

surface realization

1	2	3	4	5	6	7	8	9	10
tall	cool	smart	black	new					building
N|A	A–A	A–A	A–A	A–A	A–A	A–A	A–A	A–A	A|N

The determiner is realized from the *building* proplet in the arc which enters the noun from the verb (not shown, see TExer 2.2.9).[13]

A repetition of a functor-argument relation is subject gapping:

15.4.5 UNBOUNDED REPETITION IN SUBJECT GAPPING (TExer Sect.5.2)

Bob bought an apple, ∅ peeled a pear, ..., and ∅ ate a peach

The repeated object\predicate relations share the initial subject.

Another repetition of a functor-argument relation is predicate gapping:

15.4.6 UNBOUNDED REPET. IN PREDICATE GAPPING (TExer Sect.5.3)

Bob **bought** an apple, Jim ∅ a pear, ..., and Bill ∅ a peach**.**

The repetition is a combination of subject/predicate and object\predicate relations sharing the initial verb.

A third repetition of a functor-argument relation is object gapping:

15.4.7 UNBOUNDED REPETITION IN OBJECT GAPPING (TExer Sect.5.4)

Bob bought ∅, Jim peeled ∅, ..., and Bill ate **a peach.**

The repeated subject/predicate relations share the final object.

An example of repeating functor-argument relations at the phrasal level is based on a sequence of one prepnoun modifying the next (TExer Sect. 5.1; NLC 15.3.6):

15.4.8 UNBOUNDED REPETITION OF PREPOSITIONAL NOUN PHRASES

Fido ate the bone on the table under the tree in the garden...

There are two genuine readings: one modifies the noun bone adnominally, the other the verb eat adverbially. In both, the repetition is based on the modifier|modified relation because under the tree modifies table and in the garden modifies tree. This is in contradistinction to the coordination 15.4.4.

Repeating functor-argument relations at the clausal level may be shown by a sequence of clausal object sentences each of which takes a clausal object as its second argument except for the last one.

[13] In addition to the use of coordination in adn repetition there is noun repetition, as in Fido, Tucker, and Buster (TExer Sect. 3.6), and verb repetition as in owns, cleans, and drives (NLC Sect. 8.1).

15.4.9 UNBOUNDED REPETITION OF OBJECT CLAUSES (TExer Sect.5.5)

John says that Bill believes ... that Mary loves Tom.

We conclude with the unbounded repetition of adnominal clauses:

15.4.10 UNBOUNDED ADNOMINAL CLAUSE REPETITION (TExer Sect.5.6)

Mary saw the man who loves the woman ... who fed Fido.

The continuation solution to extrapropositional constructions in DBS is incrementing the prn value when entering a subclause and decrementing it in the corresponding exit, without any of the embedding in PSG illustrated in 15.4.1 and 15.5.1. In the donkey sentence (15.3.8), for example, the subclause is handled by the operations N↓V (TExer 3.3.12) and V↑N (TExer 3.3.15), which establish intrasentential extrapropositional clausal adnominal relations.[14]

15.5 Iteration instead of Recursion

Hauser, Chomsky, and Fitch (2002) designate recursion[15] as an essential[16] property of natural language structure. It is widely known, however, that every recursive function may be rewritten as an equivalent iterative[17] function[18] and vice versa.[19]

Computationally, the choice between iteration and recursion depends on the application and associated considerations of efficiency (Lantzman 2007). Conceptually, however, the two mechanisms are profoundly different: iteration is a flat chaining, while recursion is a repeated embedding, as in fractals.[20]

[14] For the iteration of center-embedded relative clauses in German see CLaTR 9.3.1.

[15] In computer science, recursion evolved at the end of the 1950s within the development of the programming language BASIC (Rinderknecht 2014). There are, however, earlier examples in the foundations of mathematics, such as the Ackermann function (Ackermann 1928), the Fibonacci numbers (Fibonacci 1202 A.D), and Indian Mathematics (200 BC).

[16] The alleged importance of recursion in natural language has been challenged by Everett (2005, 2013), who argues that the Pirahã language of the Amazon basin has no embedding (and a fortiori no recursion). In response, Nevins, Pesetsky, and Rodrigues (2009) reanalyzed Everett's data and concluded that Pirahã does have recursion (and a fortiori embedding). According to Stadler (2018), Pirahã had 800 speakers in the year 2018.

[17] Even the Ackermann function as a textbook example of recursion is routinely programmed as an iteration in exercises of today's computer science: https://www.inf-schule.de/

[18] As an early computer science paper on iteration, Rinderknecht (2014) cites Anonymous (1977). However, in the year 1669, iteration was used already by Newton for approximating the roots (or zeroes) of a real-valued function.

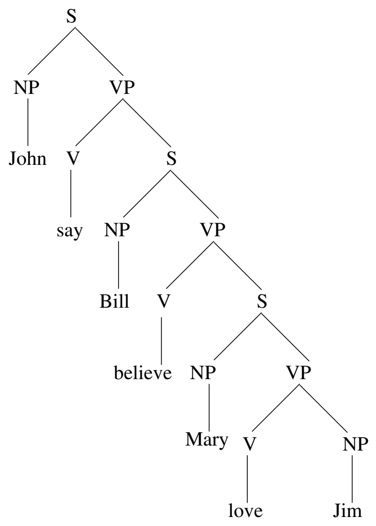
[19] The reason for Chomsky's preference of recursion over iteration may be the context-free phrase structures of Generative Grammar, which go back to the recursive rewrite rules of the Post (1936) production systems.

[20] The classic example of recursion in computer science is x = x+1. Here the '=' does not mean equality, but *is replaced by*, like the arrow in the PSG rewrite rules (15.5.2).

Intuitively, iterative chaining seems better suited for modeling the time-linear structure of natural language than recursive embedding. This is supported by the comparison of the PSG and the DBS analysis in 15.1.1. and the analyses 15.4.1 (PSG) and 15.4.4 (DBS). It may also be shown by the following example John said that Bill believes ... that Mary loves Jim, which resembles 15.1.1 except for the absent unbounded dependency.

The embedding nature of the PSG recursion on the left is made explicit by the equivalent bracketing structure on the right:[21]

15.5.1 PSG TREE AND EQUIVALENT BRACKETING STRUCTURE



S

S[NP VP]

S[NP[John] VP[V S]]]

S[NP[John] VP[V [say] S[NP VP]]]]

S[NP[John] VP[V [say] S[NP [Bill] VP [V S]]]]

S[NP[John] VP[V [say] S[NP [Bill] VP [V [believe] S [NP VP]]]]]

S[NP[John] VP[V [say] S[NP [Bill] VP [V [believe]
S [NP [Mary] VP [V NP]]]]]]

S[NP[John] VP[V [say] S[NP [Bill] VP [V [believe]
S [NP [Mary] VP [V [love] NP [Jim]]]]]]]

Both formats are substitution-driven and based on the same rewrite rules:

15.5.2 PSG GENERATING BOTH FORMATS IN 15.5.1

S[22]→ NP VP
VP→ V S
VP→ V NP

The recursion is provided by the first two rules and stopped by the third rule. The derivation is completed by applying lexical PS-rules:

[21] Bracketing may be (i) plain, (ii) structured, or (iii) labeled. The plain bracketing for the phrase structure of Fido snored is [S[NP[Fido]][VP[V[snored]]]]. It is turned into a structured bracketing by writing the next biggest constituent into a new line with indentation. A labeled bracketing is derived from a plain bracketing by using the nonterminals as subscripts to the opening brackets, i.e. $[_S[_{NP}$[Fido]]$[_{VP}[_V$[snored]]]]. For PSG tree comparison, plain bracketing is the most transparent.

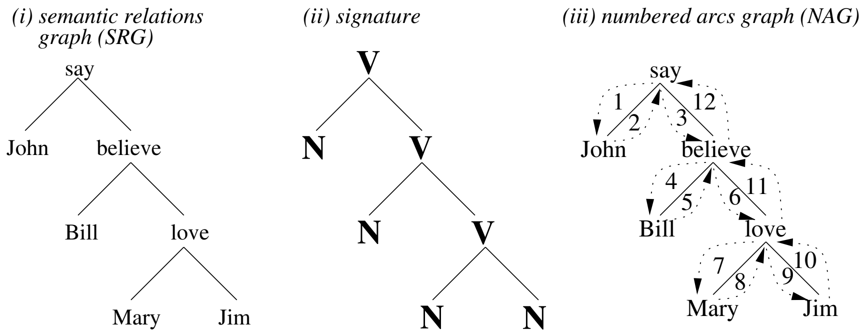
15.5.3 Lexical PSG rules which terminate recursion in 15.5.1

NP → John, Bill, Mary, Jim,...
V → say, believe, love, ...

In 15.5.1, the embedding nature of substitution in general and recursion in particular is clearly characterized by the equivalent bracketing structure shown on the right: each substitution and each new recursion cycle is grown from the inside out, using partial order.[22]

The alternative to substitution-driven "generation" is data-driven iteration, based on computing possible continuations in the hear and the speak mode:

15.5.4 Iteration of object clauses in DBS



(iv) surface realization

1	2	3	4	5	6	7	8	9	10	11	12
John	said	that	Bill	believes	that	Mary	loves	Jim			.
V/N	N/V	V\V	V/N	N/V	V\V	V/N	N/V	V\N	N\V	V\V	V\V

In contrast to PSG, there are no nonterminals like S, NP, VP, or V in DBS. Instead, DBS connects content words (or rather content proplets) directly with semantic relations of structure (Sect. 2.5), coded by address.[25]

Consider the following presentation of the example as a set of proplets. The iteration is started by the second value of [arg: [person x] (believe 28)] of say and ended by the second value of the [arg: [person y][person z]] feature of love:

---

[23] In PSG, S is the nonterminal symbol for 'Start' or 'Sentence.' In initial position, S functions like the starter of a substitution-driven motor. If there is a choice between rewrite rules, e.g. VP → V S and VP → V NP, it is theoretically at random, but in practice guided by hand. instead of a partial linear order, one could use a simultaneous (parallel) rule application, but it doesn't seem to have achieved wider acceptance.

[24] This makes a general computational pattern matching between the output of a substitution-driven language derivation and a context of use, as would be needed in 1.2.2, practically impossible (FoCL Sect. 22.2).

[25] The letters in the DBS signature, here **V** and **N**, represent proplet attributes (and not nonterminal nodes) and are reused in the names of the navigation rules, e.g. N/V.

15.5.5 John says that Bill believes that Mary loves Jim.

sur: john	sur:	sur:	sur: bill
noun: [person x]	verb: say	verb: believe	noun: [person y]
cat: snp	cat: #ns3′ #a′ decl	cat: #ns3′ #a′ v	cat: snp
sem: nm m	sem: pres ind	sem: *that* pres	sem: nm m
fnc: say	arg: [person x] (believe 28)	arg: [person y] (love 29)	fnc: believe
mdr:	mdr:	fnc: (say 27)	mdr:
nc:	nc:	mdr:	nc:
pc:	pc:	nc:	pc:
prn: 27	prn: 27	pc:	prn: 28
		prn: 28	

sur:	sur: mary	sur: jim
verb: love	noun: [person z]	noun: [person w]
cat: #ns3′ #a′ v	cat: snp	cat: snp
sem: *that* pres	sem: nm f	sem: nm m
arg: [person z] [person w]	fnc: love	fnc: love
fnc: (believe 28)	mdr:	mdr:
mdr:	nc:	nc:
nc:	pc:	pc:
pc:	prn: 29	prn: 29
prn: 29		

Coding the iteration of object clauses by means of extrapropositional continuation values defined by address establishes the iteration relation between proplets without embedding and is independent of the proplets' storage location in A-memory.

15.6 Content- instead of Coordinate-Addressable Memory

The characteristic differences between the agent- and the sign-based approach may be summarized as follows:

15.6.1 CHARACTERISTIC DIFFERENCES AT A GLANCE

1. agent-based ontology	sign-based ontology
2. data-driven	substitution-driven
3. operations coded as procedures	rules defined in a metalanguage
4. substructure by concatenation	substructure by embedding
5. repetition by iteration	repetition by recursion

For memory, however, no such alternative should be construed because there is no monitoring in sign-based systems, therefore no need for a memory, and consequently no preference for a certain kind of database. In agent-based DBS, in contrast, the memory is one of the three main components of cognition, the others being the interface and the operation component.

Among the many computational memories available today, the most basic distinction is between content-addressable and coordinate-addressable databases (Chisvin and Duckworth 1992). Which of these is suited best for building a computational cognition in a data-driven agent-based approach?

Historically, content-addressable databases came first (Buck 1955, Bachman 1973), followed by the coordinate-addressable ones (Codd 1977). Given the limitations of computational storage at the time, it was the coordinate-addressable approach which was favored by IBM for long term development.

A coordinate-addressable (also called location-addressable) memory is like a public library in which a book may be (a) permanently stored wherever there happens to be a free slot (random access) and (b) retrieved with the help of a catalog which (1) orders entries alphabetically by name of the author and (2) relates the book's title to the storage location used, e.g. X2732d.

A content-addressable memory, in contrast, is like a private library in which books with certain properties, e.g. the same general topic, the country of origin, or the color of the cover, are grouped together on certain shelves ready to be browsed without the help of a separate catalog (CLaTR Sect. 4.1). Today, content-addressable databases are used for computational applications which are huge, need to be very fast, and in which the data do not change much.

For building a computational cognition, it became apparent that coordinate-addressable databases, like a commercial RDBMS, are too unwieldy.[26] As an alternative, the schema of a classic network database seemed more promising. As work developed, the data structure of records in a classic network database was replaced by the data structure of proplets, and the database schema was adjusted to being content-addressable.

In practical use, the data stored in a DBS memory may be read and copied but never changed,[27] like sediment. The only way to correct or amplify is by adding content at the now front (6.6.6), like a diary entry, which refers to the content to be corrected by address, without touching.

In short, the memory of DBS cognition is (i) content-addressable and (ii) without any change of stored data. Correction is restricted to adding content.[28]

---

[26] When Kycia's implementation of the DBS2 grammar (NLC Chap. 13 hear mode, Chap. 14 speak mode) in Java was successfully reimplemented by Fischer (2002) as a standard RDBMS, it turned out that making small changes to the grammar, like changing a category segment, had suddenly become prohibitively laborious.

[27] This is in contradistinction to RDBMSs. Their consistency checks and associated repairs require access which may be misused for theft, fraud, and other forms of "correction" and "doctoring" (CLaTR Sect. 4.1). A recent method to impede this kind of break-in is the block chain technology, which is based on encrypting the relations within the relational database. The design of DBS also prevents any editing of stored data, but access for reading and copying must still be managed.

[28] For content-addressable databases which change data for error correction see Pontarelli and Ottavi (2013).

Summary

Database Semantics (DBS) is a data-driven agent-based theory of semantic interpretation and production. It is founded on (i) interfaces for recognition and action, (ii) an on-board orientation system, and (iii) an on-board database for the storage and retrieval of content. The theory is formalized as the integrated software design of a talking robot's cognition. Its basic standard of success is maintaining the agent in a state of balance in a changing environment containing other agents (ecological niche).

For the artificial agent, the surrounding environment has the form of raw data which are treated as given and (a) detected by the sensors and (b) produced by the actuators of the agent's interface component: in recognition, raw data are mapped into concepts; in action concepts are mapped into raw data. Artificial recognition and action are not only a practical necessity for building a computational cognition, but also indispensable theoretically for a truly grounded semantics.

The concepts serve as the core and continuation values of proplets, which are defined as non-recursive feature structures with ordered attributes. Proplets serve as the data structure of DBS and are connected into contents by the semantic relations of structure, i.e. classical functor-argument and coordination, at the elementary, phrasal, and clausal levels of grammatical complexity. Coding the relations by proplet-internal address makes concatenated proplets order-free, which is essential for storage and retrieval in the artificial agent's content-addressable database (memory).

The nonlanguage concepts of recognition and action are reused in language cognition. Conversely, the semantic relations of functor-argument and coordination of language are reused in nonlanguage cognition. This allows reconstructing the cycle of natural language communication from the hear mode to the think mode to the speak mode and back to the hear mode (turn taking). In addition there is switching between language and nonlanguage processing, and between recognition and action.

When alert, the DBS robot automatically monitors current events, including its own actions. In recognition, content is in the time-linear order of arrival and serves as the motor of data-driven DBS cognition. Think mode processing of the incoming data and resonating content in memory may result in action output. Action is realized as raw data in the time-linear order of departure.

It has been shown (TCS) that the computational complexity of a time-linear approach without recursive ambiguity (C1-LAG) is linear. Because recursive ambiguity does not occur in the DBS cycle of natural language communication, the cycle runs in linear time.

Bibliography

Ackermann, W. (1928) "Zum Hilbertschen Aufbau der reellen Zahlen," *Mathematische Annalen*, Vol. 99:118–133

Ágel, V. (2000) *Valenztheorie*. Tübingen: Gunter Narr

Aho, A.V. and J.D. Ullman (1977) *Principles of Compiler Design*, Reading, Mass.: Addison-Wesley

AIJ = Hausser, R. (2001) "Database Semantics for Natural Language," Artificial Intelligence (AIJ), Vol. 130.1, Amsterdam, Elsevier (available at `lagrammar.net`)

Andersen, P.B. (1990) *A theory of computer semiotics: semiotic approaches to construction and assessment of computer systems*, Volume 3 of Cambridge series on human-computer interaction. Cambridge University Press

Anonymous (Feb. 1977) "Depth-first digraph algorithms without recursion", *Proceedings of the Seventh International Technical Symposium on Computer Science Education*, ACM SIGCSE. Atlanta, Georgia, USA: ACM Press, pp. 151–153

Aristotle, *Analytica Priora*, in Aristoteles Latinus III.1–4, L. Minio-Paluello (ed.), Bruges–Paris: Desclée de Brouwer, 1962.

Asher, N. and J. Pelletier (2012) "More Truths about Generic Truth" University of Alberta, `https://sites.ualberta.ca/francisp/papers/AsherPelletierFinalVersion.pdf`

Atkinson, R.C., and R.M. Shiffrin (1968). Chapter: Human memory: A proposed system and its control processes. In Spence, K. W., and J.T. Spence: *The psychology of learning and motivation* (Volume 2). New York: Academic Press. pp. 89–195

Austin, J.L. ([1955]1962) *How to Do Things with Words*, Oxford: Clarendon Press

Bachman, C.W. (1973) "The Programmer as Navigator,", 1973 ACM Turing Award Lecture, *Comm. ACM,* Vol. 16:11.653–658

Bandura, A. (1986) *Social Foundations of Thought and Action: A Social Cognitive Theory,* Englewood Cliffs, N.J.: Prentice-Hall

Bandura, A. (1997) *Self-efficacy: the exercise of control*, New York: W.H. Freeman

Barcan Marcus, R. (1961) "Modalities and Intensional Languages," *Synthese*, Vol. 13.4:303–322

Bar-Hillel, Y. (1964) *Language and Information. Selected Essays on Their Theory and Application*, Reading, Mass.: Addison-Wesley

Barwise, J., and J. Perry (1983) *Situations and Attitudes*, Cambridge, Mass.: MIT Press

Barsalou, W., W.K. Simmons, A.K. Barbey, and C.D. Wilson (2003) "Grounding conceptual knowledge in modality-specific systems," TRENDS in Cognitive Sciences, Elsevier, Vol. 7.2:84–91

Barwise, J., and J. Perry (1983) *Situations and Attitudes*, Cambridge, Mass.: MIT Press

Baum, W.M. (2002) "From molecular to molar: a paradigm shift in behavior analysis," *Journal of the Experimental Analysis of Behavior*, Vol.78(1):95–116

Baum, W.M., and H.C. Rachlin (1969) "Choice as time allocation," *Journal of the Experimental Analysis of Behavior*, Vol. 12:861–874

Benson, D.F. (1994) *The Neurology of Thinking*, New York: OUP
`https://doi.org/10.1002/ana.410360535`

Bernard. C. (1865) *Introduction à l'étude de la médecine expérimentale,* first English translation by Henry Copley Greene, published by Macmillan 1927; reprinted 1949

Beutel, B. (1997) *Malaga 4.0*, CLUE-Manual, Friedrich Alexander Universität Erlangen Nürnberg

Biederman, I. (1987) "Recognition-by-components: a theory of human image understanding," *Psychological Review*, Vol. 94:115–147

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)

Boas, H.C., and I.A. Sag (eds.) (2012) *Sign-Based Construction Grammar*. Stanford: CSLI Publications. [Distributed by U. of Chicago Press]

Bloomfield, L. (1933) *Language*, New York: Holt, Rinehart, and Winston

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)

Briandais, R. de la (1959) "File Searching Using Variable Length Keys," Proc. Western Joint Computer Conf. 15, 295-298

Brinton, L.J. (2000) *The structure of modern English: a linguistic introduction.* Amsterdam: John Benjamins

Broadbent, D. (1958) *Perception and Communication*, New York: Oxford University Press

Brooks, R.A. (1986) "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation* 2.1:14–23

Buck, D.A. (1955) *Encyclopedia of Computer Science and Technology*, Vol 2:283

Cannon, W.B. (1932) *The Wisdom of the Body*, The Norton Library

Carlson, G. N. (1982) "Generic terms and generic sentences," *Journal of Philosophical Logic*, Vol. 11.2:145–181

Carnap, R. (1928) *Der logische Aufbau der Welt, 2nd ed.*, Hamburg: Felix Meiner 1961

Carnap, R. (1947) *Meaning and Necessity*, Chicago, Illinois: The University of Chicago Press

Carpenter, B. (1992) *The Logic of Typed Feature Structures*, Cambridge: CUP

Chen, Fang (2006) *Human Factors in Speech Interface Designs*, Springer

Chisvin, L., and R.J. Duckworth (1992) "Content-addressable and associative memory," p. 159–235, in M.C. Yovits (ed.), *Advances in Computer Science, 2nd ed.*, Academic Press

Chomsky, N. (1965) *Aspects of a Theory of Syntax*, Cambridge Mass.: MIT Press

Chomsky, N. (1968) "Remarks on Nominalization," Linguistics Club, Indiana University. Reprinted in: R. Jacobs and P. Rosenbaum (eds.), *Readings in English Transformational Grammar* 1979:184-221, MA: Ginn, Waltham

Chomsky, N., and M. Halle (1968) *The Sound Pattern of English*, New York: Harper Row

Church, A. (1932). "A set of postulates for the foundation of logic," *Annals of Mathematics* Series 2. 33 (2): 346–366

CLaTR = Hausser R. (2011) *Computational Linguistics and Talking Robots; Processing Content in DBS*, pp. 286. Springer (preprint 2nd ed. at `lagrammar.net`)

Codd, E.F. (1970) " A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM*, Vol.13.6: 377–387

Craik, F. I. M. and Lockhart, R. S. (1972). "Levels of processing: A framework for memory research." *Journal of Verbal Learning and Verbal Behaviour*, Vol. 11(6):671–684.

Cresswell, M. (1972) "The world is everything that is the case," *Australasian Journal of Philosophy*, Vol. 50:1-13.

Christou, S. and H. H. Bülthoff (2000) "Perception, representation and recognition: A holistic view of recognition," Spatial Vision, Vol. 13.2,3:265–275

CTGR = Hausser, R. (2017a) "A computational treatment of generalized reference," *Complex Adaptive Systems Modeling*, Vol. 5.1:1–26. Also available at `http://link.springer.com/article/10.1186/s40294-016-0042-7` and `lagrammar.net`

Darwin, C. (1859) *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life,* London: John Murray

Davidson, D. (1967) "Truth and Meaning," *Synthese*, Vol. VII:304–323

Derr, M. and K. McKeown (1984) "Using focus to generate complex and simple sentences," ACL '84 *Proceedings of the 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics*, pp. 319–326

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

Dürr, H.P. (2015) "Why physics needs ontology?" `https://www.mathematik.uni-muenchen.de/~bohmmech/BohmHome/files/duerr_ontology.pdf`

Dubnau, J., A.-S. Chiang, L. Grady, J. Barditch, S. Gossweiler, J. McNeil, P. Smith, F. Buldoc, R. Scott, U. Certa, C. Broger, T. Tully (2003) "The staufen/pumilio Pathway Is Involved in Drosophila Long-Term Memory," *Current Biology*, Vol. 13.4:286-296

Earley, J. (1970) "An Efficient Context-Free Parsing Algorithm," *Commun. ACM* 13.2:94–102, reprinted in B. Grosz, K. Sparck Jones, and B.L. Webber (eds.) 1986

Edmondson, W. (2017) *The Sequential Imperative: General Cognitive Principles and the Structure of Behaviour*, Boston: Brill

Einstein, A. (1920) *Über die spezielle und die allgemeine Relativitätstheorie*, translated by Robert W. Lawson, *Relativity: The Special and General Theory*, New York: Henry Holt, 1920; New York: Bartleby.Com, 2000. Available at `https://pdfs.semanticscholar.org/8f7e/5acbd8f5638d259d99ee664dc98c8bc98a1b.pdf`

Ekman, P. (1999) "Basic Emotions," Chapter 3 in *Handbook of cognition and emotion*, T. Dalgleish and M. Powers (eds), John Wileys

Elmasri, R., and S.B. Navathe (2017) *Fundamentals of Database Systems, 7th ed.* (1st ed. 1989), Redwood City, CA: Benjamin-Cummings

Euler, L. (1842) *Lettres de L. Euler a une Princesse D'Allemagne* [Partie ii., Lettre XXXV., ed. Cournot. – ED.] `https://archive.org/details/lettresdeleuler01courgoog/page/n9`

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

Fairbanks, A. (1898) editor and translator: *The First Philosophers of Greece*, London: K. Paul, Trench, Trubner

Fedorenko, E., and R. Varley / (2016) "Language and thought are not the same thing: evidence from neuroimaging and neurological patients," Ann. N. Y. Acad. Sci. 2016; Vol. 1369.1:132–153

Fibonacci (1202) *Liber Abaci*, translated by Sigler, L. E. (2002), *Fibonacci's Liber Abaci*, Springer

Fiengo, R., and R. May (1994) *Indices and Identity*, Cambridge, Mass: MIT Press.

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.J. (1988) "The Mechanisms of Construction Grammar" *Proceedings of the Fourteenth Annual Meeting of the Berkeley Linguistics Society (1988)*, pp. 35-55

Fischer, W. (2002) *Implementing Database Semantics as an RDBMS* (in German), Studienarbeit am Institut für Informatik der Universität Erlangen-Nürnberg (Prof. Meyer-Wegener), published as CLUE-Arbeitsbericht 7 (2004)

Flügge-Lotz, I. (1968) *Discontinuous and Optimal Control*, New York: McGraw-Hill

Flouri, T. (2012) *Pattern matching in tree structures*, Dept. of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Ph.D. dissertation

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

Fogal, D. (2016) "Review of Stephen Finlay's Confusion of Tongues: A Theory of Normative Language," in *Ethics*, Vol. 127

Fredkin, E. (1960) "Trie Memory," *Commun. ACM* Vol. 3.9:490–499

Frege, G. (1879) *Begriffsschrift. Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle: L. Nebert

Friedrich, M., I. Wilhelm, J. Born, and A.D. Friederici (2015) " Generalization of word meanings during infant sleep," *Nature Communications*, doi: 10.1038/ncomms7004. `http://www.nature.com/-ncomms/2015/150129/ncomms7004/full/ncomms7004.html` (15-01-28)

Gallup, G.G. Jr. (1970). "Chimpanzees: Self recognition," *Science* Vol. 167 (3914):86–87

Gazdar, G. (1981) "Unbounded Dependencies and Coordinate Structure," *Linguistic Inquiry* Vol. 12.2:155–184

Geach P. (1962) *Reference and Generality: An Examination of Some Medieval and Modern Theories*, Ithaca, New York: Cornell University Press

Gentner, D. (1982) "Why nouns are learned before verbs: Linguistic relativity versus natural partitioning," in S. A. Kuczaj (ed.), *Language development: Vol. 2. Language, thought and culture* (pp. 301-334). Hillsdale, NJ: Erlbaum

Gentner, D. and L. Smith (2012) "Analogical reasoning." in V. S. Ramachandran (ed.) *Encyclopedia of Human Behavior (2nd ed.)*, pp. 130–136. Oxford, UK: Elsevier

Glucksberg, S. (2001) *Understanding Figurative Language: From Metaphor to Idioms*, Oxford University Press

Greenberg, J. (1963) "Some Universals of Grammar with Particular Reference to the Order of Meaningful Elements," in J. Greenberg (ed.)

Greenberg, J. (ed.) (1963) *Universals of Language*, Cambridge, Mass.: MIT Press

Grice, P. (1975). "Logic and conversation," in Cole, P.; Morgan, J. *Syntax and semantics. 3*: Speech acts. New York: Academic Press. pp. 41–58.

Groenendijk, J., and M. Stokhof (1978) "Semantics, pragmatics and the theory of meaning," *Journal of Pragmatics*, Volume 2.1:49-70

Gualterus Burlaeus (1988) *De puritate artis logicae tractatus longior*, Hamburg: Felix Meiner Verlag

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., C. Weber, et al. (2009) "JSLIM – Computational Morphology in the Framework of the SLIM Theory of Language," *Communications in Computer and Information Science,* Vol. 41:10–27

Harbsmeier, C. (1996) "A Sinologist's Handlist of Traditional Western Rhetorical Terms," University of Oslo, version 13.10.1996, pp. 88

Harbsmeier, C. (2001) "May Fourth Linguistic Orthodoxy and Rhetoric" *New Terms for New Ideas*, pp. 373–410, ed. Lackner, Kurtz, Amelung, Leiden: Brill

Harman, G. (1963) "Generative Grammar without Transformational Rules: a Defense of Phrase Structure," *Language* Vol. 39:597–616

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. (1980) "The Place of Pragmatics in Model Theory," *Formal Methods in the Study of Language*, Amsterdam University: Mathematical Center Tracts 135 (available at `lagrammar.net`)

Hausser, R. (1999) "Database Semantics: Natural Language Interpretation, Inferencing, and Production." Proceedings of the Second International Conference on Cognitive Science, Waseda University, Tokyo, Japan (available at `lagrammar.net`)

Hausser, R. (2003) "Reconstructing Propositional Calculus in Database Semantics," in H. Kangassalo et al. (eds.), *Information Modeling and Knowledge Bases XIV,* Amsterdam: IOS Press Ohmsha (available at `lagrammar.net`)

Hausser, R. (2005) "Memory-based Pattern Completion in Database Semantics", *Language and Information*, Vol. 9.1:69–92 (available at `lagrammar.net`)

Hausser, R. (2015) "From Montague Grammar to Database Semantics," *Language and Information*, Vol 19(2):1–16 (available at `lagrammar.net`)

Hausser, R. (2016) "Computational Foundations of Database Semantics," *Language and Information*, Vol 11:99–121 (available at `lagrammar.net`)

Hawkins, B.S. (1993) "Peirce and Frege, a question unanswered," *Modern Logic*, Vol. 3.4:376–383

HTBR = Hausser, R. (2016) *How to Build a Talking Robot*, pp. 121, Springer, in print since 2016

Hinton, G.E. (1992) "How neural networks learn from experience," Scientific American, Sept. 1992, p. 145–151

Hobbs, J.R., M.E. Stickel, D. Appelt, and P. Martin (1993) "Interpretation as abduction," *Artif. Intell.* Vol. 63:69–142

Hodges, W. (2010) "Modality in mathematics," `http://wilfridhodges.co.uk/semantics12.pdf`

Kahneman, D. (1973) *Attention and Effort*, New York: Prentice Hall

Kamp, J.A.W. (1980) "A Theory of Truth and Semantic Representation," Groenendijk et al. (eds.) *Formal Methods in the Study of Language*, Amsterdam University: Mathematical Center Tracts 135

Kamp, J.A.W., and U. Reyle (1993) *From Discourse to Logic*, Parts 1 and 2, Dordrecht: Kluwer

Karray, F.; M. Alemzadeh; J.A. Saleh; M.N. Arab (2008) "Human-Computer Interaction: Overview on State of the Art," *International Journal on Smart Sensing and Intelligent Systems*, Vol. 1.1:137–159

Kass, R., and Finin, T. (1988) "Modeling the user in natural language systems," *Computational Linguistics*, Vol. 14:5–22

Katz, J., and P. Postal (1964) *An Integrated Theory of Linguistic Descriptions*, Cambridge, Mass.: MIT Press

Kempson, R., and A. Cormack (1981) "Ambiguity and quantification," *Linguistics and Philosophy* 4:259–309

Kiefer, F. (2018) "Two kinds of epistemic modality in Hungarian," in Guentchéva, Z. (ed.) *Empirical Approaches to Language Typology*, DOI: 10.1515/9783110572261–013

King, J.C., and K.S.Lewis, "Anaphora," *The Stanford Encyclopedia of Philosophy (Summer 2017 Edition),* Edward N. Zalta (ed.), URL = `https://plato.stanford.edu/archives/sum2017/entries/anaphora/`

Knuth, D.E., J.H. Morris, and V.R. Pratt (1977) "Fast Pattern Matching in Strings," SIAM Journal of Computing Vol. 6.2:323–350

Koskenniemi, K. (1983) *Two-Level Morphology*, University of Helsinki Publications No. 11

Kycia, A. (2004) *An Implementation of Database Semantics in Java*. M.A. thesis, CLUE

Lakoff, G., and S. Peters (1969) "Phrasal conjunction and symmetric predicates," in D.A. Reibel and S. Schane (eds.) *Modern Studies of English*, Englewood Cliffs, NJ: Prentice Hall. 113–142

Langacker, R. (1969) "Pronominalization and the Chain of Command," in D.A. Reibel and S. A. Shane (eds.), 160–186

Lantzman, E. (2007) "Iterative vs. Recursive Approaches," `https://www.codeproject.com/Articles/21194/Iterative-vs-Recursive-Approaches`

Lech, R. K., O. and B. Güntürkün (2016) "An interplay of fusiform gyrus and hippocampus enables prototype- and exemplar-based category learning," *Behavioural Brain Research*, Vol: 311:239–246.

Lee, Kiyong (2004) "A computational treatment of some case-related problems in Korean," *Perspectives on Korean Case and Case Marking*, 21–56, Seoul: Taehaksa

Lees, R.B. (1960) "The Grammar of English Nominalizations," *International Journal of American Linguistics*, Vol. 26.3, Part 2

Lees, R.B. and E.S. Klima (1963) "Rules for English pronominalization" *Language* Vol. 39.1:17–28

Lewis, C.I, and C.H. Cooper (1932) *Logic, Symbolic and Mathematical*, The Century co., Dover reprint 1959

Leśniewski, S. (1929) *"Grundzüge eines neuen Systems der Grundlagen der Mathematik,"* Warsaw: Fundamenta Mathematicae Vol. 14:1–81

Lewis, D. (1970) "General Semantics," *Synthese*, Vol. 22.1/2:18-67

Lerner, J.S., Ye Li, P. Valdesolo, and K.S. Kassam (2015) "Emotion and Decision Making," *Annual Review of Psychology*, Vol. 66:799–823

Lieb, H.-H. (1992) *"The Case for a new Structuralism,"* in H.-H. Lieb (ed.), Prospects for a New Structuralism. Amsterdam-Philadelpia: John Benjamins

Lorenz, O., and G. Schüller (1994) "Präsentation von LA-MORPH," *LDV-Forum*, Vol. 11.1:39–51. Reprinted in R. Hausser (ed.) 1996

Lorenz, O. (1996) *Automatische Wortformerkennung für das Deutsche im Rahmen von Malaga*, Magisterarbeit, CLUE, Friedrich Alexander Universität Erlangen Nürnberg

Marr, D. (1982) *Vision*, New York: W.H. Freeman and Company

Margulies, C., T. Tully, J. Dubnau (2005) "Deconstructing memory in Drosophila," Curr. Biol. 15: R700–R713.

Marty, A. (1908) *Untersuchungen zur Grundlegung der allgemeinen Grammatik und Sprachphilosophie*, Vol. 1, Halle: Niemeyer, reprint Hildesheim, New York: Olms (1976)

Mayring, P. (2000) "Qualitative Inhaltsanalyse," *Forum Qualitative Sozialforschung* `http://qualitative-research.net/fqs/fqs-d/2-00inhalt-d.htm` (01-11-26)

McCulloch, W., and W. Pitts (1943) "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, Vol. 5.4:115–133

McKeown, K. (1985) "Discourse Strategies for Generating Natural-Language Text," *Artificial Intelligence*, Vol. 27:1–41

McWilliams, C. (1946), *Southern California: An Island on the Land*, New York City: Duell, Sloan, and Pearce

Meadows, D.H. (2008), *Thinking in Systems : a Primer,* Vermont: Chelsea Green Publishing

Menon, V. (2011) "Developmental cognitive neuroscience of arithmetic: implications for learning and education," ZDM (Zentralblatt für Didaktik der Mathematik). Vol. 42.6:515–525

Meulen, A. ter (2012) "Generics in information structure; exceptions vs. counterexamples," *Recherches Linguistic de Vincenne*, Vol. 41:103–122

Miikkulainen, R. (1993) *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon and Memory*, Cambridge, Mass.: MIT Press.

Miller, G., E. Galanter, and K. Pribram (1960) *Plans and the structure of behavior*, New York: Holt, Rinehart and Winston

Miller, G., R. Beckwith, C. Fellbaum, D. Gross, and K. Mille (1993) `wordnetcode.princeton.edu/5papers.pdf`

Montague, R. (1970) "English as a formal language," in B. Visentini (ed.), *Linguaggi nella societa e nella tecnica,* Edizioni di Communita, pp. 188-221

Montague, R. (1973) "The Proper Treatment of Quantification in Ordinary English," in J. Hintikka, J. Moravcsik, P. Suppes (eds.) *Approaches to Natural Language*, Dordrecht, Reidel: 221–242

Montague, R. (1974) *Formal Philosophy*, Collected Papers, R. Thomason (ed.), New Haven: Yale University Press

Navon, D., and D. Gopher (1979) "On the economy of the human-processing system," *Psychological Review*, Vol. 86.3:214-255

Neumann, J.v. (1945) *First Draft of a Report on the EDVAC*, IEEE Annals of the History of Computing, 27-75

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

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

Newton, I. (1687/1999) *The Principia: Mathematical Principles of Natural Philosophy.* I.B. Cohen and A. Whitman (Translators), Berkeley: University of California Press

Nichols, J. (1992) *Linguistic Diversity in Space and Time*, The U. of Chicago Press

Nilsson, N. (1998) *Artificial Intelligence: A New Synthesis*, San Francisco: Morgan Kaufmann

NLC = Hausser R. (2006) *A Computational Model of Natural Language Commmunication: Interpretation, Inference, and Production in DBS*, pp. 360. Springer (preprint 2nd ed. at `lagrammar.net`)

Norman, D.A., and D.E. Rumelhart (1975) *Explorations in Cognition*, San Francisco: Freeman

Ogata, Katsuhiko (2010) *Modern Control Systems (Fifth ed.)*. Prentice Hall. ISBN 978-0-13-615673-4

Ogden, C.K., and I.A. Richards (1923) *The Meaning of Meaning*, London: Routledge and Kegan Paul

Owano, N. (2017) "Wait, did I just see Atlas robot do a backflip?" *Tech Xplore*, `https://techxplore.com/news/2017-11-atlas-robot-backflip.html`

Pantin, C.F.A. (1926) "On the Physiology of Amoeboid Movement–the Effect of Calcium," *Brit. Journ. Exp. Biol.*, Vol. 3:297–312

Partee, B. (1973) "Some transformational extensions of Montague grammar," *Journal of Philosophical Logic* Vol. 2:509–534

Peirce, C.S. (1903) "Lectures on Pragmatism", CP 5.171

Peirce, C.S. (1931–1935) *Collected Papers*. C. Hartshorne and P. Weiss (eds.), Cambridge, MA: Harvard Univ. Press

Pei Wang (2013) *Non-Axiomatic Logic, A Model of Intelligent Reasoning*, World Scientific Publishing, ISBN 978-981-4440-27-1

Peters, S., and R. Ritchie (1973) "On the generative power of transformational grammar," *Information and Control* Vol. 18:483–501

Peters, S., and D. Westersthål (2006) *Quantifiers in Language and Logic*, Oxford: Oxford Univ. Press

Phillips, S. and W.H. Wilson (2010) "Categorial Compositionality: A Category Theory Explanation for the Systematicity of Human Cognition," *PLoS Computational Biology* Vol. 6:7

Pollard, C., and I. Sag (1987): Information-based Syntax and Semantics. Volume 1: Fundamentals. Stanford: CSLI Publications

Pollard, C., and I. Sag (1994): Head-Driven Phrase Structure Grammar. Chicago: University of Chicago Press

Pontarelli, S., and M. Ottavi (2013) "Error Detection and Correction in Content Addressable Memories by Using Bloom Filters" *IEEE Transactions on Computers archive*, Vol. 62.6: 1111–1126

Post, E. (1936) "Finite Combinatory Processes — Formulation I," *Journal of Symbolic Logic*, Vol. I:103–105

Primus, B. (2012) *Semantische Rollen*, Universitäts Verlag Winter Heidelberg, KEGLI Band 12

Putten, S. van (2019) "Perception verbs and the conceptualization of the senses: The case of Avatime," Accepted for publication in *Linguistics*

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, Mass.: MIT Press

Read, S. (2017) "Aristotle's Theory of the Assertoric Syllogism," `https://www.st-andrews.ac.uk/~slr/The_Syllogism.pdf`

Reimer, M., and E. Michaelson (2014). "Reference," The Stanford Encyclopediaof Philosophy(Winter 2014 Edition), Edward N. Zalta (ed.), `http://plato.stanford.edu/archives/win2014/entries/reference/`

Rescher, N. (1969) *Many-valued Logic*, New York: McGraw-Hill

Ricoeur, P. (1975) *The Rule of Metaphor: Multi-Disciplinary Studies in the Creation of Meaning in Language*, trans. Robert Czerny with Kathleen McLaughlin and John Costello, S. J., London: Routledge and Kegan Paul 1978. (Toronto: University of Toronto Press 1977)

Rinderknecht, C. (2014) *A Survey on Teaching and Learning Recursive Programming*, Department of Programming Languages and Compilers, Eötvös Loránd University, Budapest, Hungary. http://crinderknecht.free.fr/pub/ie2014.pdf

Rosch, E. (1975) "Cognitive representations of semantic categories," *J. of Experimental Psychology*, General 104:192–253

Ross, J.R. (1969) "On the cyclic nature of English pronominalization," in D.A. Reibel and S.A. Schane (eds.), p. 187–200

Russell, B. (1905) "On denoting," *Mind* 14:479–493

Sag, I. (2012) "Sign-Based Construction Grammar: An Informal Synopsis," in Boas and Sag (eds.), pp. 69–202

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

Schegloff, E.A. (2000). "Overlapping talk and the organization of turn-taking for conversation," Language in Society. 29 (1): 1–63. doi:10.1017/s0047404500001019

Searle, J.R. (1969) *Speech Acts*, Cambridge, UK: Cambridge University Press

Shannon, C.E., and W. Weaver (1949) *The Mathematical Theory of Communication*, Urbana, Illinois: University of Illinois Press

Simon, R.M. (2017a) "Want a Robot to Walk Like You? Don't Expect It to Look Human," *Wired*, https://www.wired.com/story/want-a-robot-to-walk-like-you-dont-expect-it-to-look-human/

Simon, M. (2017b) "Boston Dynamics' Atlas Robot Does Backflips Now and It's Full-Tilt Insane" *Wired*, https://www.wired.com/story/atlas-robot-does-backflips-now/

Smolensky, P. (1991), "Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems," in Hinton ed. (1991), *Connectionist Symbol Processing*, p. 159–216, Cambridge Mass: MIT Press

Sowa, J.F., and A.K. Majumdar / (2003) "Analogical Reasoning," A. Aldo, W. Lex, and B. Ganter, eds. (2003) *Conceptual Structures for Knowledge Creation and Communication*, LNAI 2746:16–36, Springer

Spranger, M., M. Loetzsch, and S. Pauw (2010) "Open-ended Grounded Semantics" https://csl.sony.fr/wp-content/themes/sony/uploads/pdf/spranger-10b.pdf

Stadler, J. (2018) "Daniel Everett: Wie ein Missionar zum Atheisten wurde," *PROFIL.at*

Stangl, W. (2007) "Inhaltsanalyse, content analysis," http://arbeitsblaetter.stangl-taller.at/FORSCHUNGSMETHODEN/Inhaltsanalyse.shtml (07-09-19)

Steels, L. (2008) "The symbol grounding problem has been solved. so what's next?" in *Symbols and Embodiment: Debates on Meaning and Cogni-tion*, ed., M. de Vega, Oxford: Oxford University Press

Tarski, A. (1935) "Der Wahrheitsbegriff in den Formalisierten Sprachen," *Studia Philosophica*, Vol. I:262–405

Tarski, A. (1944) "The Semantic Concept of Truth," *Philosophy and Phenomenological Research*, Vol. 4:341–375

TCS = Hausser R. (1992) "Complexity in Left-Associative Grammar," *Theoretical Computer Science*, Vol. 106.2:283-308 (available at lagrammar.net)

TExer = Hausser, R. (2017c) *Twentyfour Exercises in Linguistic Analysis, DBS software design for the Hear and the Speak mode of a Talking Robot*, pp. 324. DOI: 10.13140/RG.2.2.13035.39200 (available at lagrammar.net)

Thagard, P. (2006) "Evaluating explanations in law, science, and everyday life," *Association of Psychological Science*, Vol. 15.3:141–145

Thalheim, B. (2000) *Entity-Relationship Modeling*, Springer

Thomason, R. (ed.) (1974) *Formal Philosophy. Selected Papers of Richard Montague*, New Haven: Yale University Press

Tooby, J., and L. Cosmides (1990) "The past explains the present: emotional adaptations and the structure of ancestral environments," *Ethology and Sociobiology*, Vol. 11:375–424

Trier, J. (1928) *Der deutsche Wortschatz im Sinnbezirk des Verstandes. Von den Anfängen bis zum Beginn des 13. Jahrhunderts*. Heidelberg 1973: Winter, ISBN 3-533-00535-6 (Habilitationsschrift, Universität Marburg 1928).

Ullmann, S. (1957) *The Principles of Semantics*, Oxford: Blackwell

Venn, J. (1881) *Symbolic Logic*, London: MacMillan

Weyhrauch, R. (1980) "Prolegomena to a Formal Theory of Mechanized Formal Reasoning," *Artificial Intelligence*, Vol. 13.1–2:133–170, DOI:10.1016/0004-3702(80)90015-6

Wiener, N. (1948) *Cybernetics: Or the Control and Communication in the Animal and the Machine*, Cambridge, MA: MIT Press

Wilson, E.O. (1998) *Consilience: The Unity of Knowledge*, New York: Alfred Knopf

Winograd, T. (1972) *Understandmg Natural Language*, New York: Academic Press

Zadeh, L. (1965) "Fuzzy sets," *Information and Control*, Vol. 8:338–353

Zentall, T. R., E. A. Wasserman, O. F. Lazareva, R. K. R. Thompson, and M. J. Rattermann (2008) "Concept learning in animals," *Comparative Cognition and Behavior Reviews*, Vol. 3:13–45

Zhang Yinsheng and Qiao Xiaodong (2009) "A Formal System of Aristotelian Syllogism Based on Automata Grammar," *Computer Science and Information Engineering*, 2009 WRI World Congress

Name Index

Subject Index

This book is about changing from the sign-based ontology of language analysis to the agent-based ontology of language communication in particular and cognition in general.

Some of the Topics Addressed

Computer Science

suitable data structure
algorithm computing continuations
computational complexity
content-addressable database schema
processing medium and processing modalities
mapping raw data into content (recognition)
mapping content into raw data (action)
content Mechanisms: matching, pointing, baptism
iteration vs. recursion
string search and trie structure

Analytic Philosophy

agent-based vs. sign-based ontology
Content kinds: concept, indexical, name
Semantic kinds: referent, property, relation
type-token distinction
context of interpretation
generalized reference
inferencing with and without set theory
symbolic logic and the classical syllogisms
polysemy-homonymy

Linguistics

procedural vs. metalanguage-based semantics
speak, think, and hear mode in the cycle of communication
semantic relations of structure coded by address
time-linear derivation order
levels of grammatical complexity
literal and nonliteral use
on-board orientation system and the interpretation of indexicals
monitoring current recognition and action
limiting current processing to the now front
loomlike clearance of the now front