The Ontology of Communication

data-driven agent-based or substitution-driven sign-based?

NOV. 2019–JUNE 2020

ROLAND HAUSSER

Friedrich Alexander Universität Erlangen Nürnberg (em.)

July 15, 2020

email: rrh@linguistik.uni-erlangen.de

河宇士

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

Preface

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 Today, the ontology of physics is based on a space-time continuum, protons, electrons, neutrons, quarks, neutrinos, etc.

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 the computational cognition of a talking robot?[1]

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] First released under the title Eight Easy Pieces in June 2020, this small volume combines eight papers written between October 2019 and June 2020. Their common theme is how to adjust today's language sciences to the age of computers and artificial intelligence. For a systematic presentation in book form see Computational Cognition (CC).

All papers have been rechecked and conservatively revised, cleaning up the prose, correcting irregularities in some examples, and adjusting page breaks to the present format. Notable improvements may be found in Sects. 3.1, 3.2, 5.3, 6.3, 6.6, 7.4.

Contents

1. Laboratory Set-Up of Database Semantics

June 12. 2020

Abstract

The analysis of natural language in today's linguistics, analytic philosophy, and computer science is either (i) data-driven agent-based or (ii) substitution-driven sign-based. A sign-based ontology has the apparent advantage that it obviates any need for an interface component with sensors for vision and audition, and actuators for manipulation and vocalization. In an age when artificial vision, audition, manipulation, locomotion, and computers did not exist, this was a necessity. The question is: how to adjust today's language research to the age of computers and artificial intelligence by changing to a data-driven agent-based ontology?

keywords: time-linearity, type transparency, computational complexity, successful communication, content, declarative specification, operational implementation, fragment of a natural language, upscaling cycles

1.1 Early Times

The absence of computers did not stop the grammarians of the ancient and recent past from contributing essential notions representing important insights, such as accusative, active, adjective, agglutination, agreement, allomorph, analytic, aorist, clause, comparation, conjunction, dative, determiner, domain, event, function, future, genitive, imperfect, inflection, isolating, medium, modifier, morpheme, morphology, nominative, noun, object, passive, perfect, phrase, pragmatics, predicate, pronoun, proposition, range, relation, semantics, subclause, subject, syntactic mood, syntax, synthetic, tense, and verbal mood. Without these notions most of modern linguistics would be unthinkable.

A recent attempt to bring language science into modern times was Chomsky's generative grammar for characterizing the innate universal structure of natural language (nativism). Rewrite rules generate constituent structures from the S node (for sentence or start) by repeated substitution, resulting in phrase structure trees which are defined in terms of "dominance" and "precedence." By adding a transformation component to a context-free phrase structure base, the computational complexity of Transformational Grammar increased from polynomial to undecidable (Peters and Ritchie 1973).

Chomsky emphasized repeatedly that generative grammar was "not intended" for modeling communication: "To avoid what has been a continuing misunderstanding, it is perhaps worth while to reiterate that a generative grammar is not a model for a speaker or a hearer." (Chomsky 1965, p. 9). Yet, as shown by the external analogy with anatomy, it is unlikely that a supposedly innate universal model of natural language would be without a speak mode, a hear mode, and a transfer channel, especially in language acquisition.

In consequence, a majority of the field moved from nativism to empirical studies based on large data and statistics. For building a talking robot, this is too imprecise. In analogy, 'If the Martians came to Earth and modeled cars statistically they would never run.' Instead, the Martians would have to chose a car in good running condition, take it apart, study the parts and the functional flow, and reconstruct the mechanisms of the motor, the wheels, the breaks, the transmission, etc., until the reassembled vehicle would run again as before.

1.2 Study of the Language Signs

A truly classic pioneer of modern linguistics was the Swiss linguist de Saussure (1857–1913), who formulated the most important properties of the natural language signs as the *premiere princip*, l'arbitraire de signe,[1] and the *second principe*, caractère linéaire du signifiant.[2] These principles are as valid today as when they were first proposed.

Regarding the second principle, de Saussure continues in good humor:

> Ce principe est évident, mais il semble qu'on ait toujours négligé de l'énoncer, sans doute parce qu'on l'a trouvé trop simple; cependent il est fondamental et les conséquences en sont incalculables; son importance est égale à celle de la première loi. Tout le mécanisme de la langue en dépend.[3]

Ignoring time-linearity is one of those aberrations which are so frequent in the history of science and which often take several centuries to be rectified.

The first attempt at combining time-linearity with detailed grammatical analysis and efficient computation was NEWCAT'86. Still without a distinction between the speak and the hear mode, it presents time-linear derivations for 221 examples of German and 114 examples of English, programmed in Lisp and published with the source code.

1.2.1 NEWCAT-STYLE PARSE OF Fido dug the bone up. (COL 3.3.4)

```
* (z Fido dug the bone up \.)

    Linear Analysis:

    *START
    1
       (N-H) FIDO
       (N A UP V) DUG
    *NOM+FVERB
    2
       (A UP V) FIDO DUG
       (GQ) THE
    *FVERB+MAIN
    3
```

```
   (GQ UP V) FIDO DUG THE
   (S-H) BONE
*DET+NOUN
4
   (UP V) FIDO DUG THE BONE
   (UP NP) UP
*FVERB+MAIN
5
   (V) FIDO DUG THE BONE UP
   (V DECL) .
*CMPLT
6
   (DECL) FIDO DUG THE BONE UP .
```

The grammatical analysis is a formatted *trace* of the computational operations. Each numbered derivation step begins with the rule name, and combines a sentence start and a next word. The result is shown as the input to the next derivation step. As a direct reflection of the computational application of the grammar rules, tracing is the ultimate form of 'type transparency' (Berwick and Weinberg 1984). Computational tracing as the exclusive method of grammatical analysis is used in all subsequent work of what became DBS.

Like NEWCAT'86, *Computation of Language* (CoL 1989) is still sign-based, but expands the time-linear NEWCAT approach to computational complexity analysis. For example, the formal language $a^k b^k c^k$ is context-sensitive in the PSG hierarchy and parses in exponential time, but a C1 language in the LAG hierarchy and parses in linear time[4] (CoL 6.4.3, FoCL 10.2.2):

1.2.2 LA Grammar for $a^k b^k c^k$

$LX =_{def} \{[a\ (a)], [b\ (b)], [c\ (c)]\}$
$ST_s =_{def} \{[(a)\ \{r_1, r_2\}]\}$
r_1: (X) (a) $\Rightarrow$ (aX) $\{r_1, r_2\}$
r_2: (aX) (b) $\Rightarrow$ (Xb) $\{r_2, r_3\}$
r_3: (bX) (c) $\Rightarrow$ (X) $\{r_3\}$
$ST_F =_{def} \{[\varepsilon\ rp_3]\}$.

[1] 'First principle: arbitrariness of the sign.' It refers to the fact that different languages may use different surfaces, e.g., fauteuil, sessel, and poltrona, for the same kind of thing, here 'easy chair,' based on different *conventions* within the different language communities.

[2] 'Second principle: linear character of the sign.' It refers to the fact that language signs follow each other in a certain grammatical order. Changing the order results in a change of meaning or in ungrammaticality.

[3] 'This principle is obvious, but it seems that stating it explicitly has always been neglected, doubtlessly because it is considered too elementary. It is, however, a fundamental principle and its consequences are incalculable. Its importance equals that of the first law. All the mechanisms of language depend on it.' De Saussure [1916](1972), p. 103.

[4] The term 'time-linear' refers to a grammatical derivation order while the term 'linear time' refers to a computational complexity degree.

A lexical entry like [a(a)] in the set LX consists of a surface, here a, and a category, here (a). The set ST_s happens to contain only one start state, namely {[(a) {r_1, r_2}]}; this means that the first input must have the category (a), i.e. it must have the surface a, and that the rules applying to the first and the second input are limited by the rule package to r_1 and r_2. Rule r_1 adds an (a), r_2 subtracts an (a) and adds a (b), while r_3 subtracts a (b) from the category.

The rule package of r_1 is {r_1, r_2}, i.e. after r_1 has applied, r_1 and r_2 are tried on the next word, and accordingly for the rules packages of r_2 and r_3. The set ST_F contains only one final state, namely {[ε rp_3]}, i.e. the category must be empty (ε) and the currently activated rule package must be that of r_3.

Compared to the context-sensitive PSG (FoCL 8.3.7), the LAG is exceedingly plain. Furthermore, the LA Grammars for context-free $a^k b^k$ (CoL 10.2.3) and for context-sensitive $a^k b^k c^k$ are in the same language class of DBS and the number of coefficients, as in $a^k b^k c^k d^k$, $a^k b^k c^k d^k e^k$, etc., has no effect on the linear complexity of their LA Grammars. Like the natural language analysis 1.2.1, an $a^k b^k c^k$ expression is analyzed as a formatted trace of the parse:

1.2.3 Sample derivation of **aaabbbccc** with active rule counter

```
* (z a a a b b b c c c)
;  1: Applying rules (RULE-1 RULE-2)
;  2: Applying rules (RULE-1 RULE-2)
;  3: Applying rules (RULE-1 RULE-2)
;  4: Applying rules (RULE-2 RULE-3)
;  5: Applying rules (RULE-2 RULE-3)
;  6: Applying rules (RULE-2 RULE-3)
;  7: Applying rules (RULE-3)
;  8: Applying rules (RULE-3)
; Number of rule applications: 14.

    *START-0
    1
       A (A)
       A (A)
    *RULE-1
    2
       A A (A A)
       A (A)
    *RULE-1
    3
       A A A (A A A)
       B (B)
    *RULE-2
    4
       A A A B (A A B)
       B (B)
    *RULE-2
    5
       A A A B B (A B B)
```

```
   B (B)
*RULE-2
6
   A A A B B B (B B B)
   C (C)
*RULE-3
7
   A A A B B B C (B B)
   C (C)
*RULE-3
8
   A A A B B B C C (B)
   C (C)
*RULE-3
9
   A A A B B B C C C (NIL)
```

Expressions which are not in the language, e.g. aaabbc, are analyzed to the point of the ungrammatical continuation, here aaabb+c, and rejected as such. While PSG derivations are substitution-driven by always starting with the same S symbol followed by successive applications of rewrite rules (computing possible substitutions), LAG derivations are data-driven by processing the input surface from beginning to end (computing possible continuations). The LAG hierarchy is the first, and so far the only, complexity hierarchy which is orthogonal to the PSG hierarchy (TCS).

1.3 Using Successful Communication for the Laboratory Set-Up

In face to face dialogue, the hearer's interpretation begins with the speaker's first word. From there, the hearer follows the sequence of incoming surfaces incrementally, with the speaker at least one word ahead. In indirect communication based on writing or recorded message, the speaker's lead is unlimited.

This could be taken as a reason for starting the scientific analysis of natural language communication with the speak mode. However, there is a more important aspect to the distinction between the two modes, namely the difference in the respective input and output: the speak mode takes a cognitive content as input and produces an external surface as output, while the hear mode takes an external surface as input and produces a cognitive content as output.[5]

For the scientific investigation of natural language communication, the hear mode has the practical aspect of concretely given external input, i.e. the raw data of the language-dependent surfaces. They have no meaning or grammatical properties whatsoever, but they are measurable by natural science and

[5] Software which runs "forward" for content input and "backward" for surface input (or vice versa) is not feasible. DBS inferencing, in contrast, allows *inductive* (forward) and *abductive* (backward) use because both directions take the same kind of input and produce the same kind of output.

interpretable by automatic speech recognition or optical character recognition (OCR). The input to the speak mode, in contrast, is agent-internal cognitive content which can only be inferred.

Therefore DBS starts the computational reconstruction of natural language communication with the hear mode's first step, namely the recognition of the raw surface input by means of pattern matching. The output of the hear mode is an agent-internal, purely cognitive structure: it derives the literal meaning$_1$ (PoP-1, FoCL 4.3.3) of the input surface as an agent-internal content.

In order for communication to be successful, the speaker's input content and the hearer's output content must be the same:

1.3.1 Minimal condition for communication to be successful

> In order for communication to be successful, the meaning$_1$ content derived as output in the hear mode must be the same as the meaning$_1$ content used as input by the speak mode.

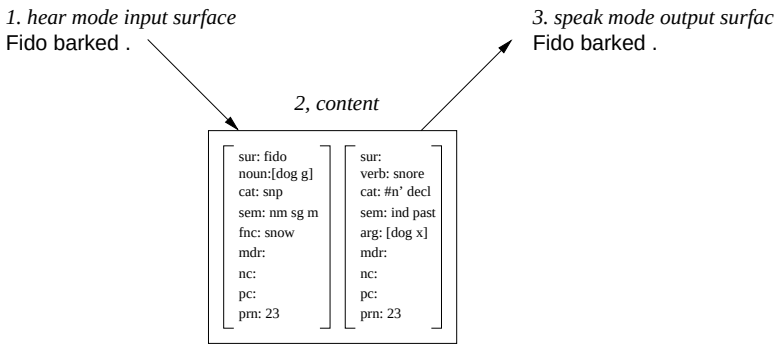
This criterion is the pivot of the DBS *laboratory set-up*:

1.3.2 Definition of the DBS laboratory set-up

- The content automatically derived as output in the hear mode is reused systematically as the input to the automatic speak mode derivation.
- The content of a given example surface is correct if, and only if, the hear mode's input surface equals the speak mode's output surface.

The laboratory set-up provides a fully automatic, clear and simple method of verification. It requires that (i) the grammatical details of the speak mode suffice for the associated hear mode to automatically derive the speaker's content and (ii) that the grammatical details of hear mode content suffice for the associated speak mode derivation to automatically produce the hearer's surface.

1.3.3 Laboratory set-up: from hear mode to speak mode



The DBS laboratory set-up produces semantically well-motivated content as input for the speak mode. It is the first and so far the only method of linguistic analysis by which the hear and the speak mode benefit each other.

The laboratory set-up is based on switching off inferencing, temporarily limiting the think-speak mode to traversing meaning$_1$ content and producing literal surface representations in the natural language of choice ('narrative speak mode'). This resembles a sign-based approach in that it excludes pragmatics, but differs in that it has an explicit notion of content and includes the reconstruction of the speak and the hear mode. When the direction from speaker to hearer outside the laboratory set-up is re-established and inferencing is switched back on, the speak mode may realize inference content as language-dependent surfaces and the hear mode may interpret the surfaces as inference content – data-driven, without any need for additional software.

1.4 From Operational Implementation to Declarative Specification

NEWCAT and CoL take a complete expression as input and process it symbol by symbol in left-associative[6] order. FoCL, in contrast, supplies (i) individual 'next words' separately by automatic word form recognition, (ii) intertwines each hear mode operation application with a next word look-up, and (iii) defines the operations to integrate the 'next word' into the current 'sentence start.' This came with a change from the ordered triple analysis of a word form in NEWCAT to the proplet format as a nonrecursive feature structure with ordered attributes, serving as the computational data structure.

For example, the ordered triple analysis of dug in 1.2.1 was changed into the following proplet:

1.4.1 Transition from an ordered triple to a lexical proplet

ordered triple format	*proplet format of DBS*
[dug (N A up V) dig]	sur: dug verb: dig cat: N′ A′ up′ V sem: *up* ind past arg: mdr: nc: pc: prn:

The two formats differ as follows.

[6] Aho and Ullmann 1977, p. 47.

1.4.2 Comparing the NEWCAT-CoL approach with DBS

1. The ordered triple format does not distinguish between valency slots and valency fillers, whereas DBS proplets mark valency positions with $'$, e.g. N$'$.
2. In the ordered triple format, valency positions are canceled by deletion (as in Categorial Grammar), whereas the DBS hear mode cancels valency positions by #-marking, thus preserving the information for the speak mode.
3. Derivations in the ordered triple format prefer ending on empty category and use the complete derivation as the resulting content. A derivation in the proplet format, in contrast, results in a derivation-independent content, defined as a set of proplets connected by address.
4. The proplet format enables string-search-based storage in and retrieval from a content-addressable database contained by a cognitive agent with an interface component (agent-based ontology).
5. The on-board database interacts with the agent's interface component for the recognition and production of language surfaces, as well as the recognition of and action with nonlanguage contents.
6. The agent's on-board orientation system called STAR is used linguistically for the interpretation of the sign kind 'indexical.'

Compare the following DBS hear mode derivation with the NECAT derivation 1.2.1:

1.4.3 DBS hear mode derivation of Fido dug the bone up.

unanalyzed surface

Fido dug the bone up •

automatic word form recognition

sur: Fido	sur: dug	sur: the	sur: bone	sur: up	sur: •
noun: [dog x]	verb: dig	noun: n_1	noun: bone	noun: n_2	verb: v_1
cat: snp	cat: n' a' bp' v	cat: nn' np	cat: sn	cat: adnv	cat: v' decl
sem: nm m	sem: past	sem: def	sem: sg	sem: *up*	sem:
fnc:	arg:	fnc:	fnc:	mdd:	mdd:
prn:	prn:	prn:	prn:	prn:	prn:

syntactic−semantic parsing

1	sur: Fido noun: [dog x] cat: snp sem: nm m fnc: prn: 21	sur: dug verb: dig cat: n' a' bp' v sem: past arg: prn:		*cross−copying*
2	sur: fido noun: [dog x] cat: snp sem: nm m fnc: dig prn: 21	sur: verb: dig cat: #n' a' bp' v sem: past arg: [dog x] prn: 21	sur: the noun: n_1 cat: nn' np sem: def fnc: prn:	*cross−copying*
3	sur: fido noun: [dog x] cat: snp sem: nm m fnc: dig prn: 21	sur: verb: dig cat: #n' #a' bp' v sem: past arg: [dog x] n_1 prn: 21	sur: noun: n_1 cat: nn' np sem: def fnc: dig prn: 21	sur: bone ~~noun:~~ bone ~~cat:~~ sn ~~sem:~~ sg fnc: prn:	*absorption with simultaneous substitution*
4	sur: fido noun: [dog x] cat: snp sem: nm m fnc: dig prn: 21	sur: verb: dig cat: #n' #a' bp' v sem: ~~past~~ arg: [dog x] bone prn: 21	sur: noun: bone ~~cat: nn' np~~ ~~sem: def~~ fnc: dig prn: 21	sur: up noun: n_2 ~~cat:~~ adnv ~~sem: *up*~~ mdd: prn:	*absorption*

5	sur: fido noun: [dog x] cat: snp sem: nm m fnc: dig prn: 21	sur: verb: dig cat: #n' #a' #bp' v sem: *up* past arg: [dog x] bone prn: 21	sur: noun: bone ~~cat: nn' np~~ sem: def fnc: dig prn: 21	sur: • verb: v_1 ~~cat: v'~~ decl sem: fnc: prn:	*absorption*

result

sur: fido noun: [dog x] cat: snp sem: nm m fnc: dig prn: 21	sur: verb: dig cat: #n' #a' #bp' decl sem: *up* past arg: [dog x] bone prn: 21	sur: noun: bone cat: nn' np sem: def fnc: dig prn: 21

The bare preposition up is absorbed in line 4. It #-cancels the valency position bp′ in the third cat slot of *dig* and writes its sem value up into the initial sem slot of the verb, making it available for the speak mode.

Like the NEWCAT-style derivation 1.2.1, the derivation exactly mirrors the sequence of operation applications.[7] However, instead of taking the whole surface as input, there is incremental next word lookup, which is essential for the transition from one sentence to the next, as in a text (TExer3, Sect. 2.1). Instead of deleting valency positions, they are #-canceled in the verb's cat slot. Instead of using the complete derivation as the grammatical analysis, there results a content which is independent of the hear mode derivation and defined as a set of proplets connected by address.

1.5 Formal Fragments of Natural Language

There are formal language analyses in the tradition of symbolic logic (Montague 1974) and computational complexity (FoCL) theory which use explicit rule systems to analyze-generate limited 'fragments' of natural or formal languages like $a^k b^k$, $a^k b^k c^k$, $a^k b^k c^k d^k$, etc. (1.2.3). A fragment is precisely defined as a set of examples for the analysis of specific, natural or artificial, grammatical structures. The language data in a fragment are limited, but their analysis is required to be complete.

The use of software in the computational analysis of fragments opens a new dichotomy as compared to precomputational linguistics, namely between (i) the *declarative specification* and (ii) the *operational implementation*. The declarative specification represents the necessary properties of the software and is simultaneously suitable (a) for reading by humans and (b) for a straightforward translation into a general purpose programming language of choice. The operational implementation, in contrast, has additional accidental proper-

[7] For the sequence of explicit hear and speak mode operation applications see TExer3, Sect. 4.3.

ties, namely those which distinguish equivalent implementations in different programming languages.

After working on implementing a fragment of natural language in a general purpose programming language of choice, there naturally arises a scientific interest in leaving the accidental properties behind and work out the necessary ones in the systematic format of a declarative specification.[8] Conversely, after working on a declarative specification for a fragment of a natural language, there naturally arises a scientific interest in verifying the fragment in the form of an operational implementation.[9]

1.6 Incremental Upscaling Cycles

Once a current fragment has been supplied with a declarative specification for the speak and the hear mode and been verified by an operational implementation, the next upscaling cycle is started by extending the current fragment with a limited number of additional examples which have new syntactic and semantic properties. For this kind of work, a standard computer of today is sufficient. It provides the keyboard for input and the screen for output, which allows to implement the hear mode, the content-addressable database with its now front mechanism, and the think-speak mode navigation with and without surface realization, using placeholders for concepts.

Conclusion

For building a talking robot, the recognition and action hardware of the interface component should be co-developed with the cognition software. This holds specifically for building the on-board orientation system and supplying the placeholder values for concepts with procedural implementations (CC Chapter 11).

[8] The first operational implementations were published as NEWCAT and CoL. Work on the declarative specification began with FoCL, continued with NLC2 and CLaTR2, and culminated for now in CC and TExer3.

[9] Very useful as interesting thesis topics for students.

Bibliography

Aho, A.V. and J.D. Ullman (1977) *Principles of Compiler Design*, Reading, Mass.: Addison-Wesley

Berwick, R.C., and A.S. Weinberg (1984) *The Grammatical Basis of Linguistic Performance: Language Use and Acquisition*, Cambridge, Mass.: MIT Press

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

Chomsky, N. (1965) *Aspects of the Theory of Syntax*, Cambridge, Mass.: MIT Press

CoL = Hausser, R. (1989) *Computation of Language*, Springer

FoCL[1,2,3] = Hausser, R. (1999, 2002, 2014) *Foundations of Computational Linguistics*, Springer

Montague, R. (1974) *Formal Philosophy*, Collected Papers, R. Thomason (ed.), New Haven: Yale University Press

Peters, S., and R. Ritchie (1973) "On the Generative Power of Transformational Grammar," *Information and Control*, Vol. 18:483–501

NEWCAT = Hausser, R. (1986) *NEWCAT: Parsing Natural Language Using Left-Associative Grammar*, LNCS 231, Springer

Saussure, F. de [1916](1972) *Cours de linguistique générale*, Édition critique préparée par Tullio de Mauro, Paris: Éditions Payot

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

Tesnière, L. (1959) *Éléments de syntaxe structurale*, Paris: Editions Klincksieck

TExer3 = Hausser, R. (2020) *Twentyfour Exercises in Linguistic Analysis, DBS software design for the Hear and the Speak mode of a Talking Robot* (3rd ed. `lagrammar.net`)

2. Outline of DBS

Nov. 3, 2019

Abstract

DBS is an agent-based data-driven model of cognition. It has (i) an interface component for interacting with the environment in recognition and action, (ii) an on-board database for storing and retrieving content provided by recognition and inferencing, (iii) a now front as the arena for processing current content, (iv) an on-board orientation system (STAR), and (v) an operations component for (a) navigating and inferencing in the think mode, (b) surface-content mapping in the hear mode, and (c) content-surface mapping in the speak mode.

The central notion of DBS is cognitive content, defined as a set (order-free) of nonrecursive feature structures with ordered attributes, called proplets. As the computational data structure of DBS, proplets are completely self-contained but connected into content with the classical semantic relations of structure, i.e. functor-argument and coordination, coded by address.

Content defined as order-free but connected proplets is well-suited for efficient storage in, and retrieval from, the agent's content-addressable database, using string-search. It also supports the data-driven connecting of lexical proplets into content (nonlanguage recognition and hear mode interpretation) and the autonomous navigation along the semantic relation between proplets for activating a content, deriving new content from given content by inferencing, and mapping activated content into language-dependent surfaces.

keywords: data structure, data base schema, pattern matching, turn taking, type-token, grounding, sensory and processing media and modalities, incremental left-associative transfer in communication

2.1 Building Content in the Agent's Hear Mode

DBS defines a content in terms of concepts like square (2.7.1) or blue (2.7.3). Supplied by the agent's memory, concepts types are used to recognize raw data by matching, resulting in tokens.[1] In action, a type is adapted to a purpose as a token which is realized as raw data (2.7.2, 2.7.4).

For concatenation, the concepts are embedded as core values into nonrecursive feature structures with ordered attributes, called proplets. Proplets serve as the computational data structure of DBS. The proplets of an elementary proposition are connected by a shared proposition (prn) value and by bidirectional pointering between core and continuation values. The connections between proplets are the classical semantic relations of structure, i.e. subject/predicate, object\predicate, modifier|modified, and conjunct−conjunct. Consider the following example:

[1] The type-token distinction was introduced by C. S. Peirce (CP 4:537).

2.1.1 THE CONTENT OF The dog snored.

sur:	sur:
noun: **dog**	verb: **snore**
cat: def sg	cat: #n′ decl
sem:	sem: past ind
fnc: **snore**	arg: **dog**
mdr:	mdr:
nc:	nc:
pc:	pc:
prn: 24	prn:24

This subject/predicate relation uses the core value **dog** of the first proplet as the continuation value of the second, and the core value **snore** as the continuation value of the first (bidirectional pointering). The proplets of a content are order-free in that they are stored and retrieved according to the alphabetical sequence of their core values, but are connected by the address of their continuation values, here (snore 24) and (dog 24).

In the hear mode, the content 2.1.1 results from the following derivation:

2.1.2 SURFACE COMPOSITIONAL TIME-LINEAR HEAR MODE DERIVATION

unanalyzed surface

The dog snored •

automatic word form recognition (lexical lookup)

sur: The	sur: dog	sur: snored	sur: •
noun: n_1	noun: dog	verb: snore	verb: v_1
cat: sn' snp	cat: sn	cat: n' v	cat: v' decl
sem: def	sem: sg	sem: past ind	sem:
fnc:	fnc:	arg:	arg:
prn:	prn:	prn:	prn:

syntactic−semantic parsing

1 sur: The / noun: n_1 / cat: sn' snp / sem: def / fnc: / prn: 24 ← sur: dog / ~~noun:~~ dog / cat: sn / sem: sg / fnc: / prn: — *absorption*

2 sur: / noun: dog / cat: snp / sem: def sg / fnc: / prn: 24 ⇄ sur: snored / ~~verb:~~ snore / cat: n' v / sem: past ind / arg: / prn: — *cross−copying*

3 sur: / noun: dog / cat: snp / sem: def sg / fnc: snore / prn: 24 ; sur: / verb: snore / cat: #n' v / sem: past ind / arg: dog / prn: 24 ← sur: • / verb: v_1 / ~~cat: v'~~ decl / sem: / arg: / prn: — *absorption*

result

sur:	sur:
noun: dog	verb: snore
cat: snp	cat: #n' decl
sem: def sg	sem: past ind
fnc: snore	arg: dog
prn: 24	prn: 24

DBS hear mode derivations are time-linear and surface compositional.

Operations consist of (i) an antecedent, (ii) a connective, and (iii) a consequent. Defined as proplet patterns, operations are data-driven in that they are activated by content proplets matching an input pattern. The operations of the hear mode take two input proplets and produce one or two output proplets. The operations of the think-speak mode take one input proplet and produce one output proplet.

In the hear mode, there are three kinds of operations: (1) cross-copying with the connective $\times$, (2) absorption with the connective $\cup$, and (3) suspension with the connective $\sim$. Cross-copying encodes the semantic relations of structure such as SBJ$\times$PRED (line 2). Absorption combines function words with content word such as DET$\cup$CN (line 1). Suspension such as ADV$\sim$NOM applies when no semantic relation exists for connecting the next word with the content processed so far, as in Yesterday $\sim$ Mary danced[2] (called a discontinuity in linguistics).

Consider the hear mode operation SBJ$\times$PRED as it applies in line 2 of 2.1.2:

2.1.3 SBJ$\times$PRED APPLYING BY CROSS-COPYING

$$
\begin{array}{c}
\textit{pattern level} \\
\\
\textit{content level}
\end{array}
\begin{array}{c}
\begin{bmatrix} \text{sur:} \\ \text{noun: } \alpha \\ \text{cat: def sg} \\ \text{sem:} \\ \text{fnc:} \\ \text{prn: K} \end{bmatrix}
\begin{bmatrix} \text{sur:} \\ \text{verb: } \beta \\ \text{cat: n}' \text{ decl} \\ \text{sem: past ind} \\ \text{arg:} \\ \text{prn:} \end{bmatrix}
\Rightarrow
\begin{bmatrix} \text{sur:} \\ \text{noun: } \alpha \\ \text{cat: def sg} \\ \text{sem:} \\ \text{fnc: } \beta \\ \text{prn: K} \end{bmatrix}
\begin{bmatrix} \text{sur:} \\ \text{verb: } \beta \\ \text{cat: \#n}' \text{ decl} \\ \text{sem: past ind} \\ \text{arg: } \alpha \\ \text{prn:K} \end{bmatrix} \\
\Updownarrow \qquad \Uparrow \qquad \Downarrow \qquad \Downarrow \\
\begin{bmatrix} \text{sur:} \\ \text{noun: } \mathbf{dog} \\ \text{cat: def sg} \\ \text{sem:} \\ \text{fnc:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 24} \end{bmatrix}
\begin{bmatrix} \text{sur:} \\ \text{verb: } \mathbf{snore} \\ \text{cat: n}' \text{ decl} \\ \text{sem: past ind} \\ \text{arg:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix}
\begin{bmatrix} \text{sur:} \\ \text{noun: } \mathbf{dog} \\ \text{cat: def sg} \\ \text{sem:} \\ \text{fnc: } \mathbf{snore} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 24} \end{bmatrix}
\begin{bmatrix} \text{sur:} \\ \text{verb: } \mathbf{snore} \\ \text{cat: \#n}' \text{ decl} \\ \text{sem: past ind} \\ \text{arg: } \mathbf{dog} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:24} \end{bmatrix}
\end{array}
$$

In the hear mode, the second content proplet, here *snore*, is provided by automatic word form recognition and called the 'trigger proplet.' By matching ($\Uparrow$) the second input pattern, called the 'trigger pattern,' the operation is activated to look for a content proplet matching its first input pattern ($\Updownarrow$) at the now front (2.2.2), here the special case of a sentence start consisting of a single lexical proplet. By binding α to dog and β to snore, the consequent produces the output as content proplets ($\Downarrow$).

[2] In the initial combination, Yesterday cannot connect with danced because of intervening Mary.

2.2 Storage and Retrieval of Content in the On-Board Memory

As sets, the proplets of a content are order-free, which is essential for their storage in, and retrieval from, the agent's A-memory (formerly called word bank). The database schema of A-memory is defined as follows:

2.2.1 Two-dimensional database schema of A-memory

- *horizontal lines of tokens*
  Horizontally, proplets with the same core value are stored in the same token line in the time-linear order of their arrival.
- *vertical column of token lines*
  Vertically, token lines are in the alphabetical order induced by the letter sequence of their shared core value.

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

2.2.2 A-memory before incremental storage at the now front

(i) member proplets	*(ii) now front*	*(iii) owner*
...		
... [sur: / noun: dog / ... / prn: 3] [sur: / noun: dog / ... / prn: 6]		dog
...		
... [sur: / noun: snore / ... / prn: 5] [sur: / noun: snore / ... / prn: 7]		snore
...		

The owners equal the core values and are used for access in storage and retrieval. Proplets provided by current recognition, A-memory, or inferencing are stored at the now front in the token line corresponding to their core value:

2.2.3 Storage of 2.1.1 at the now front of A-memory

(i) member proplets	*(ii) now front*	*(iii) owner*
...		
... [sur: / noun: dog / ... / prn: 3] [sur: / noun: dog / ... / prn: 6]	[sur: chien / noun: dog / ... / prn: 14]	dog
...		
... [sur: / noun: snore / / prn: 5] [sur: / noun: snore / ... / prn: 7]	[sur: ronflait / noun: snore / ... / prn: 14]	snore
...		

Once the input proplets have been assembled into a proposition, the now front is cleared by moving it and the owners to the right into fresh memory space

(loom-like clearance). This leaves the proplets of the current content behind in what is becoming their permanent storage location as member proplets, never to be changed, like sediment, but available for being read and copied.

2.2.4 A-MEMORY AFTER NOW FRONT CLEARANCE

(i) member proplets	*(ii) now front*	*(iii) owner*
... [sur: / noun: dog / ... / prn: 3] [sur: / noun: dog / ... / prn: 6] [sur: / noun: dog / ... / prn: 14]		dog
... [sur: / noun: snore / ... / prn: 5] [sur: / noun: snore / ... / prn: 7] [sur: / noun: snore / ... / prn: 14]		snore

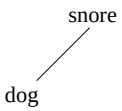
The current now front is cleared when its proplets have ceased to be candidates for additional concatenations. This is basically the case when an elementary proposition is completed (formally indicated by the automatic incrementation of the prn value for the next proposition). Exceptions are extrapropositional (i) coordination and (ii) functor-argument. In either case, 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 in the strictly time-linear derivation order.

2.3 Speak Mode Riding Piggyback on the Think Mode

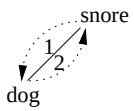
The speak mode counterpart to the graphical hear mode derivation is a presentation of the semantic relations of structure. The hear mode example 2.1.2 has only one semantic relation of structure, namely N/V for subject/predicate:

2.3.1 SEMANTIC RELATIONS GRAPH UNDERLYING THE CONTENT 2.1.1

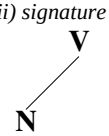
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	
The_dog	snored_	.
V/N	N/V	

The static aspects of the semantic relations of structure are shown on the left: the *(i) SRG* is based on the core values of the content and the *(ii) signature* on

the core attributes. The dynamic aspects of a think-speak mode activation are shown on the right: the arc numbers of the *(iii) NAG* are used for specifying a think mode navigation;[3] the *(iv) surface realization* shows the language-dependent production with the speak mode riding piggy-back on the think mode navigation.

The think mode operations driving the traversal of the NAG in 2.3.1 are V↙N and N↗V, and apply as follows (shown with English surface realization):

2.3.2 NAVIGATING WITH V↙N FROM *snore* TO *dog* (arc 1)

$$\begin{array}{ll} & \mathbf{V{\swarrow}N} \\ \textit{pattern level} & \begin{bmatrix} \text{verb: } \alpha \\ \text{arg: } \beta \text{ X} \\ \text{prn: K} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{sur: lexnoun}(\hat{\beta}) \\ \text{noun: } \beta \\ \text{fnc: } \alpha \\ \text{prn: K} \end{bmatrix} \\ & \Uparrow \qquad\qquad \Downarrow \\ \textit{content level} & \begin{bmatrix} \text{sur:} \\ \text{verb: snore} \\ \text{cat: \#n}' \text{ decl} \\ \text{sem: past ind} \\ \text{arg: dog} \\ \ldots \\ \text{prn: 14} \end{bmatrix} \quad \begin{bmatrix} \text{sur: The dog} \\ \text{noun: dog} \\ \text{cat: snp} \\ \text{sem: def m} \\ \text{fnc: snore} \\ \ldots \\ \text{prn: 14} \end{bmatrix} \end{array}$$

2.3.3 NAVIGATING WITH N↗V FROM *dog* BACK TO *snore* (arc 2)

$$\begin{array}{ll} & \mathbf{N{\nearrow}V} \\ \textit{pattern level} & \begin{bmatrix} \text{noun: } \beta \\ \text{fnc: } \alpha \\ \text{mdr: Z} \\ \text{prn: K} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{sur: lexverb}(\hat{\alpha}) \\ \text{verb: } \alpha \\ \text{arg: \#}\beta \text{ Y} \\ \text{prn: K} \end{bmatrix} \\ & \text{Z is NIL, or elementary and \#-marked} \\ & \Uparrow \qquad\qquad \Downarrow \\ \textit{content level} & \begin{bmatrix} \text{sur} \\ \text{noun: dog} \\ \text{cat: snp} \\ \text{sem: def m} \\ \text{fnc: snore} \\ \ldots \\ \text{prn: 14} \end{bmatrix} \quad \begin{bmatrix} \text{sur: snored\_}\bullet \\ \text{verb: snore} \\ \text{cat: \#n}' \text{ decl} \\ \text{sem: past ind} \\ \text{arg: \#dog} \\ \ldots \\ \text{prn: 14} \end{bmatrix} \end{array}$$

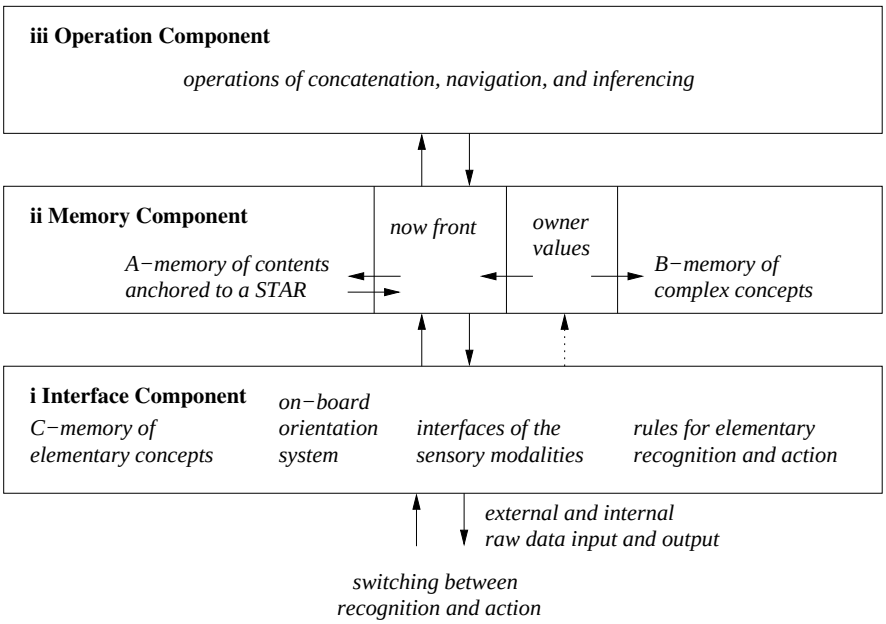
If the lexnoun rules in the sur slot of the output patterns are switched on (as assumed in the surface realization of 2.3.1), they produce a language-dependent surface using the core, cat, and sem values of the output proplet.

2.4 Graphical Summary of the DBS Component Structure

A DBS agent's component structure may be summarized as the following graph:

[3] The declarative specification of the concepts *snore* and *dog* is provided by the agent's memory and their operational implementation by the interface component (CC Chap. 11).

2.4.1 Two-dimensional layout of DBS cognition components



Cognitive content is processed at the now front. It gets proplets (a) from the interface component (aided by the owners) and (b) from A-memory. For processing, the now front provides proplets as input to (iii) the operations, which either replace the input with their output or add their output to the input. As the now front is cleared in regular intervals by moving into fresh memory space (Sect. 2.2), the processed proplets are left behind in A-memory like sediment (loom-like clearance). Processing may also result in blueprints for action, which may be copied to the interface component for realization.

2.5 Sensory Media, Processing Media, and their Modalities

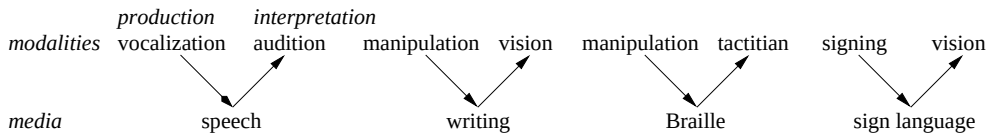
The functional equivalence required between the artificial agent and its natural prototype is defined at a level of abstraction which is above the distinction between different processing media, such as natural, mechanical, and electronic processing. Functional equivalence is shown, for example, by the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

[3] Multiple adverbial modifiers at the elementary level are usually not coordinated, but modify the verb directly (TExer Sect. 3.1).

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

In addition to the processing media there are the sensory media. In natural language communication, there exist four, each of which has two sensory modalities.[5] 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:

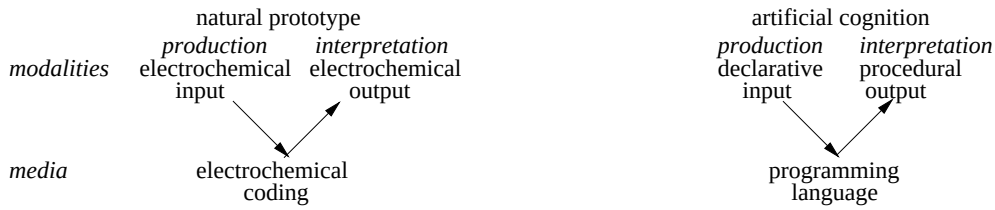
2.5.1 Sensory media and their modalities in natural language



In terms of human evolution, the primary sensory medium is speech.

While the sensory media must be the same in the natural prototype and the artificial counterpart, as required by functional equivalence, the processing media are fundamentally different between the two. For the natural prototype, neurology suggests an electrochemical processing medium, though much is unknown.[6] In artificial DBS cognition, in contrast, 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.

2.5.2 Processing media and their dual processing 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.

[5] 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.

[6] For an early overview see Benson (1994), still unsurpassed.

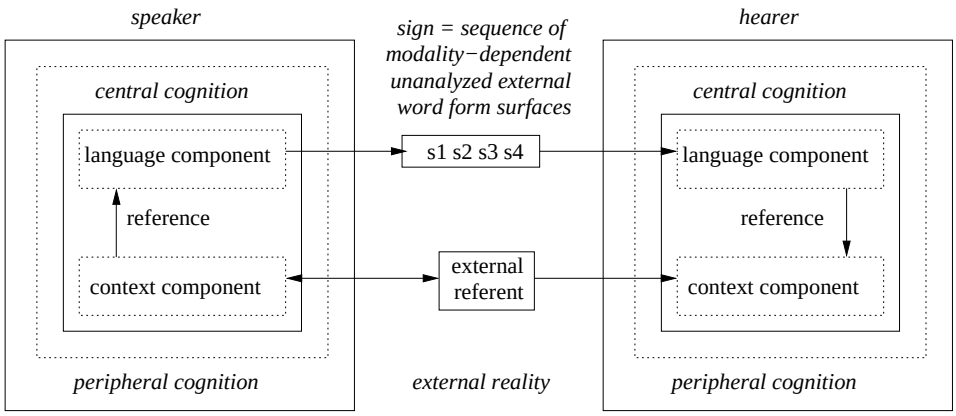
2.6 Reference as a Purely Cognitive Process

Analytic philosophy defines reference as an agent-external relation between the language sign (referring part) and the rest of the world (referred-to part).[7] Data-driven agent-based DBS treats reference (i) as an agent-internal cognitive process, (ii) confines reference to nouns (CC 1.5.3, 12.3.3), and (iii) distinguishes (1) between referring nouns with and without external surfaces and (2) between referred-to nouns with and without external[8] counterparts.

The two distinctions may be characterized by the binary features [±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 [−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, beginning with the [+surface +external] constellation between speaker and hearer:

2.6.1 Immediate reference in language communication



Agent-externally, language surfaces (shown here as s1 s2 s3 s4) 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 whatsoever, but may be measured by the natural sciences and recognized by

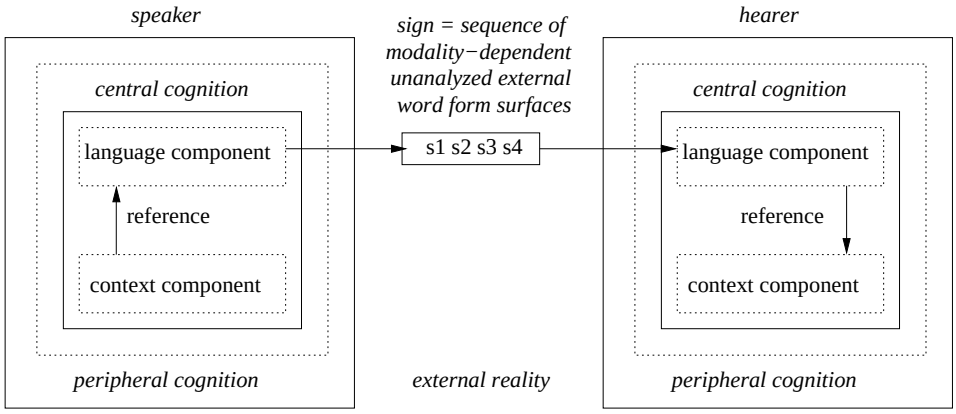
[7] Reimer and Michaelson (2014) extend the referring part from language to "representational tokens," which include cave paintings, pantomime, photographs, videos, etc. DBS goes further by generalizing the referring part to content *per se*, i.e. without the need for any external representation (2.6.3).

[8] Newell and Simon call the agent's external surroundings the *task environment* (Newell and Simon 1972).

automatic speech recognition, optical character recognition (OCR), etc. There is absolutely no external link between the external sign and the external referent, which would be a methodologically completely unacceptable *reification*.

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

2.6.2 Mediated reference in language communication



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 in immediate and mediated language reference are the same.[9]

The graphs 2.6.1 and 2.6.2 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 s4 arrives at the hearer first, which would be functionally incorrect.

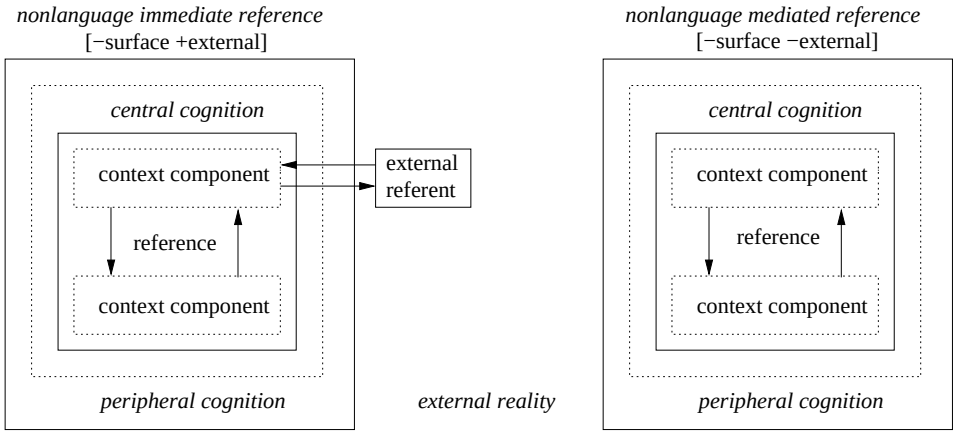
It is a pseudo-problem, however, which vanishes if each surface is transmitted individually and placed to the right of its predecessor, i.e. (((s1 s2) s3) s4). This *left-associative*[10] departure and arrival structure is the structural means to implement a time-linear derivation order in the speak and the hear mode. It allows incremental surface by surface processing, provided the derivation order is based on computing possible continuations, as in Left-Associative Grammar.

[9] On the phone, the speaker may use an immediate reference which is mediated 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.

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

Nonlanguage reference is not provided for in analytic philosophy, but exists and has the variants [−surface +external], i.e. nonlanguage immediate reference, and [−surface −external], i.e. nonlanguage mediated reference:

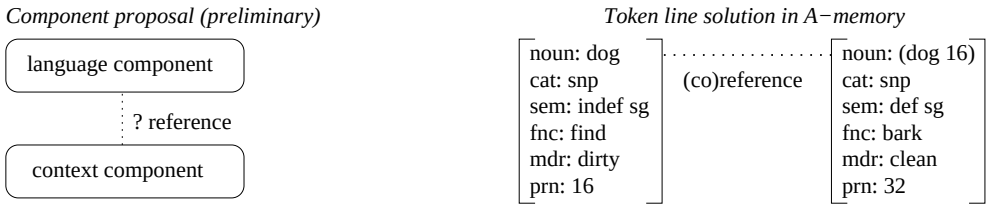
2.6.3 Nonlanguage immediate vs. mediated reference



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 (CC Sects. 3.2, 3.3) in memory.

Computationally, the conceptual view of reference as a vertical interaction between two separate cognitive components in 2.6.1–2.6.3 is implemented as a horizontal relation between two proplets in the same token line:

2.6.4 Naive vs. computational treatment of reference

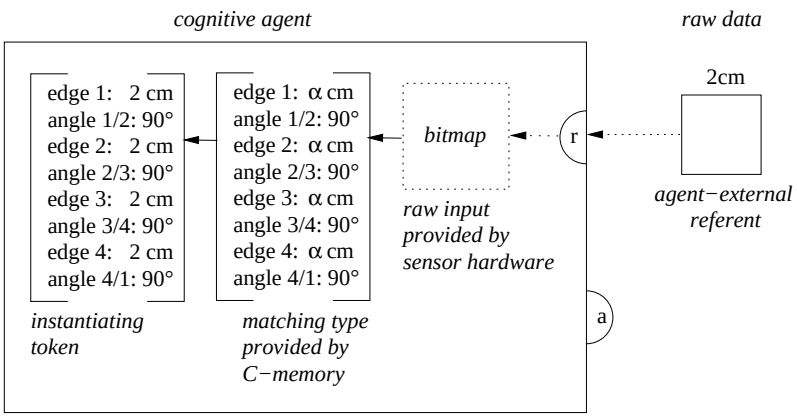


As one of the three Semantic kinds of natural language, *referent* is restricted to the Syntactic kind noun, in contradistinction to the Semantic kind *property* which is restricted to one-place verbs and non-clausal modifiers, and *relation* which is restricted to two- and three-place verbs (CC Sects. 1.5 and 6.4). The core value of a co-referring noun (shadow, copy) at the now front is always an address. The core value of the referred-to noun (referent, original) is never an address. The fnc and mdr values are free (identity in change, CC 6.4.7).

2.7 Grounding

The semantics of DBS is grounded (Barsalou et al. 2003, Steels 2008, Spranger et al. 2010). In recognition, concept types, supplied by the agent's memory, are matched with raw data, provided by sensors of the interface component:

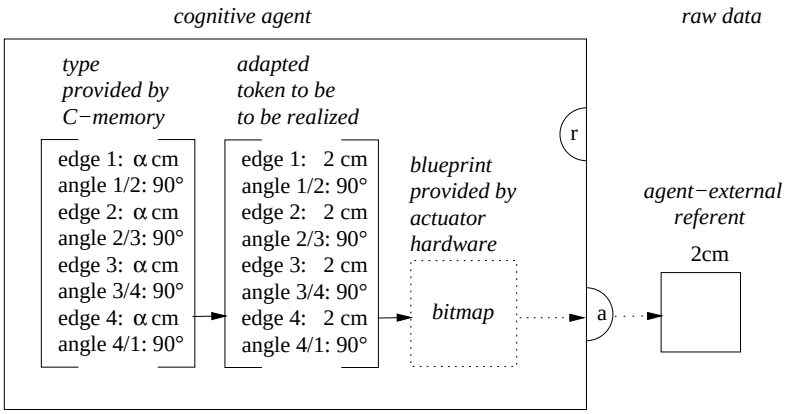
2.7.1 Recognition of square



The matching between the type and the raw data input results in an instantiating token. It equals the type except that the variables are replaced by constants resulting from appropriate measurements for additional accidental properties.

In action, a type is adapted to a token for the purpose at hand and realized by the agent's actuators as raw data:

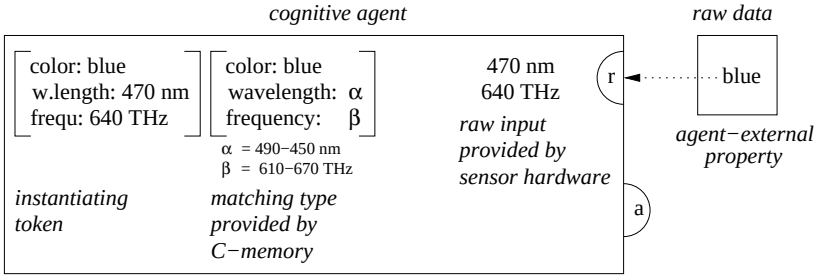
2.7.2 Action of realizing square



The token is used as a blueprint for action, (e.g. drawing a square).

Next consider the recognition of a color, here blue:

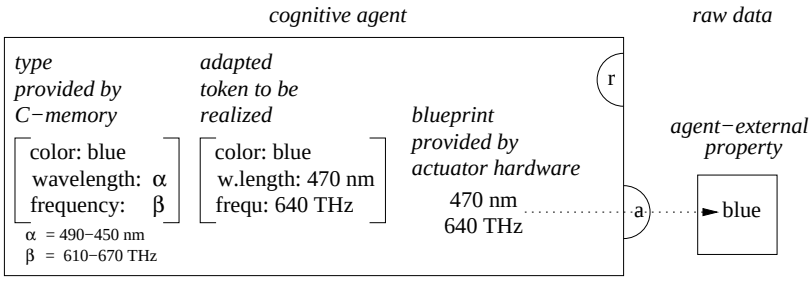
2.7.3 Recognition of blue



The principle for color recognition and for two-dimensional geometry is methodologically the same.

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

2.7.4 Action of realizing blue



The concept type matches different shades of blue. An action token results from instantiating the variables α and β as constants. The recognition and action of the color blue is a general mechanism which may be applied to all colors. It may be expanded to infrared and ultraviolet, and to varying intensity.[11]

Conclusion

Pattern matching based on the type-token relation applies to nonlanguage items (e.g. 2.7.1, 2.7.2, 2.7.3, 2.7.4) and language surfaces alike. For example, in the surfaces of spoken language the type generalizes over different pitch, timbre, dialect, and speaker-dependent pronunciation. In written language,

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

the type generalizes over the size, color, and font of the letters. Computational type-token matching is descriptively more adequate than the nonbivalent (Rescher 1969; FoCL Chap. 20.5) and fuzzy (Zadeh 1965) logics for treating vagueness because type-token matching treats the phenomenon at the root (best candidate principle in pattern matching, FoCL Sect. 5.2) instead of tinkering with the truth tables of Propositional Calculus.

Bibliography

Aho, A.V. and J.D. Ullman (1977) *Principles of Compiler Design*, Reading, Mass.: Addison-Wesley

Barcan Marcus, R. (1961) "Modalities and intensional languages," *Synthese*, Vol. 13.4:303–322

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

Benson, D.F. (1994) *The Neurology of Thinking*, New York: OUP `https://doi.org/10.1002/ana.410360535`

Biederman, I. (1987) "Recognition-by-components: a theory of human image understanding," *Psychological Review*, Vol. 94:115–147

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

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

Elmasri, R., and S.B. Navathe (2017) *Fundamentals of Database Systems, 7th ed.* (1st ed. 1989), Redwood City, CA: Benjamin-Cummings

Kiefer, F. (2018) "Two kinds of epistemic modality in Hungarian," in Guentchéva, Z. (ed.) *Empirical Approaches to Language Typology*, DOI: 10.1515/9783110572261–013

Menon, V. (2011) "Developmental cognitive neuroscience of arithmetic: implications for learning and education," ZDM (Zentralblatt für Didaktik der Mathematik). Vol. 42.6:515–525

Neumann, J.v. (1945) *First Draft of a Report on the EDVAC*, IEEE Annals of the History of Computing, 27-75

Newell, A., and H.A. Simon (1972) *Human Problem Solving*, Englewood Cliffs, New Jersey: Prentice-Hall

Peirce, C.S. (1931–1935) *Collected Papers*. C. Hartshorne and P. Weiss (eds.), Cambridge, MA: Harvard Univ. Press

Putten, S. van (2019) "Perception verbs and the conceptualization of the senses: The case of Avatime," Accepted for publication in *Linguistics*

Quine, W.v.O. (1960) *Word and Object*, Cambridge, Mass.: MIT Press

Reimer, M., and E. Michaelson (2014). "Reference," The Stanford Encyclopedia of Philosophy(Winter 2014 Edition), Edward N. Zalta (ed.), `http://plato.stanford.edu/archives/win2014/entries/reference/`

Rescher, N. (1969) *Many-Valued Logic*, New York: McGraw-Hill

Rosch, E. (1975) "Cognitive representations of semantic categories," *J. of Experimental Psychology*, General 104:192–253

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`

Steels, L. (2008) "The symbol grounding problem has been solved. so what's next?" in *Symbols and Embodiment: Debates on Meaning and Cognition*, ed., M. de Vega, Oxford: Oxford University Press

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

Zadeh, L. (1965) "Fuzzy sets," *Information and Control*, Vol. 8:338–353

3. Grounding of Concept, Indexical, and Name

Nov. 3, 2019

Abstract

The semantics of agent-based DBS is 'grounded' in that the Content kinds *concept, indexical*, and *name* have their foundation in the agent's recognition and action, whereby each Content kind has its own computational Mechanism. For a concept it is pattern *matching* between the type provided by memory and raw data provided by the agent's interface component. For an indexical it is *pointing* at a STAR value of the agent's on-board orientation system. For a name it is the address of the 'named referent' which is inserted as the core value into a lexical name proplet in an act of *baptism* .

Orthogonal to the Content kinds and their computational Mechanisms are the Semantic kinds *referent, property,* and *relation* with their associated Syntactic kind *noun, adj* and *intransitive verb*, and *transitive verb*. It is shown that the Semantic kind of *referent* is restricted to the Syntactic kind of noun, but utilizes the computational Mechanisms of matching, pointing, and baptism. Furthermore, figurative use is restricted to the computational Mechanism of matching, but uses the Semantic kinds referent, property, and relation.

keywords: Syntactic kinds of noun, adj, verb; Semantic kinds of referent, property, relation; Content kinds of concept, indexical, name; computational Mechanisms of matching, pointing, baptism; recognition vs. action; type vs. token; speak mode vs. hear mode; language vs. nonlanguage content

3.1 Apparent Terminological Redundancy

The notions noun, verb, and adjective from linguistics (philology) have counterparts in analytic philosophy, namely referent, relation, and property, and in symbolic logic, namely argument, functor, and modifier:

3.1.1 Three times three related notions

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

We take it that these variants 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:

3.1.2 1ST CORRELATION: SEMANTIC AND SYNTACTIC KIND

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

The Semantic kinds *referent, property*, and *relation* correspond to argument, 1-place functor, and 2- or 3-place functor, respectively, in Symbolic Logic. Syntactically, *property* splits up into adn, adv, prepnoun, and 1-place verb. *Relation* splits up into 2- and 3-place verbs.

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

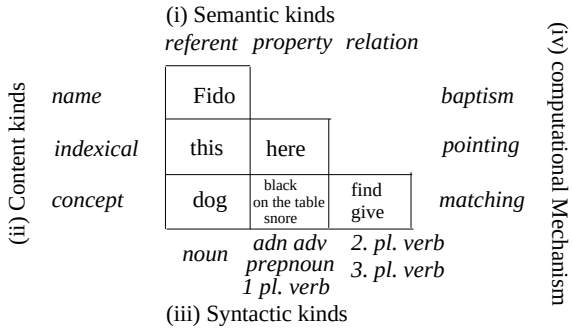
3.1.3 2ND CORRELATION: CONTENT KIND AND COMPUT. MECHANISM

	Content kind	*computational Mechanism*
a.	concept	matching
b.	indexical	pointing
c.	name	baptism

The terms of the three Content kinds and the correlated Mechanisms have had informal use in the literature,[1] but without an agent-based ontology. The essential points of the Mechanisms in DBS are their obvious computational realizations (Sects. 3.4–3.6), which have not been utilized until now.

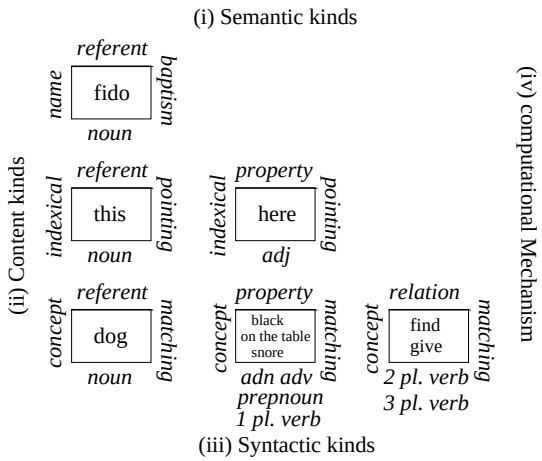
The dichotomies 3.1.2 and 3.1.3 provide 12 ($2\times2\times3$) basic notions. Empirically, they combine into six classes of proplets which constitute the semantic building blocks of DBS cognition in general and natural language communication in particular. The six classes form what we call the *cognitive square*[2]:

3.1.4 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:

3.1.5 Closer view of the Cognitive Square



The surfaces inside the rectangles have the following proplet definitions:

3.1.6 Proplets instantiating the cognitive square of DBS

referent

sur:	Fido
noun:	(dog x)
cat:	snp
sem:	nm m sg
fnc:	
mdr:	
nc:	
pc:	
prn:	

name *baptism*

noun

referent *property*

sur:		sur:	
noun:	this	adj:	here
cat:	snp	cat:	adnv
sem:	-hum sg	sem:	loc
fnc:		fnc:	
mdr:		mod:	
nc:		nc:	
pc:		pc:	
prn:		prn:	

indexical *pointing*

noun *adj*

---

[1] Examples of precomputational uses are (i) matching for concepts but without the type-token relation and its computational implementation based on content and pattern proplets, (ii) pointing for indexicals but without an on-board orientation system (STAR), and (iii) baptism but without the named referent as the core value for use in the speak and the hear mode.

[2] 'Triangle' would be appropriate as well, but the term "cognitive triangle" is already used by the cognitive behavioral therapists (CBT). Earlier it was used by Ogden&Richards (1923) for their "Semiotic Triangle." The term 'square' is well suited to express the orthogonal relation between the Syntactic_kinds-Semantic_kinds and the Content_kinds-computational_Mechanisms.

	referent	*property*			*relation*		
	sur:	sur:	sur:	sur:	sur:	sur:	
	noun: dog	adj: black	noun: table	verb: snore	verb: find	verb: give	
	cat: snp	cat: adnv	cat: adnv snp	cat: n-s3′ v	cat: n-s3′ a′ v	cat: n-s3′ d′ a′ v	
	sem: def sg	sem: pad	sem: *on* def sg	sem: pres	sem: pres	sem: pres	
concept	fnc:	fnc:	fnc:	arg:	arg:	arg:	*matching*
	mdr:	mod:	mdr:	mdr:	mdr:	mdr:	
	nc:	nc:	nc:	nc:	nc:	nc:	
	pc:	pc:	pc:	pc:	pc:	pc:	
	prn:	prn:	prn:	prn:	prn:	prn:	
	noun	*adj*	*prepnoun*	*verb* 1-pl	*verb* 2-pl	*verb* 3-pl	

In a proplet, the Semantic kind *referent* is limited to the core attribute noun, *property* is limited to the core 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 kinds *name, indexical*, and *concept* are specified by the core value of a proplet. In names, the corresponding computational Mechanism of *baptism* is implemented by inserting a 'named referent' as core value into a lexical proplet, in indexicals by *pointing* at a STAR value of the onboard orientation system, and in concepts by computational type-token *matching*

The cognitive square of DBS is empirically important because (i) figurative use is restricted to concepts, i.e. the bottom row in 3.1.4–3.1.6, 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.

3.2 Restriction of Figurative Use to Concepts

To show the restriction of figurative use to the Content kind concept let us go systematically through the three Semantic kinds:

3.2.1 Three Content kinds for the Semantic kind *REFERENT*

concept	*indexical*	*name*
sur:	sur:	sur: tom
noun: animal	noun: pro2	noun: [person x]
cat: sn	cat: sp2	cat: snp
...	...	...

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

Next consider the Semantic kind *property*, which occurs as the Content kinds (i) *concept* and (ii) *indexical*, but not as *name*. Property proplets of the content kind concept may have the core attributes adj, noun, or verb if it is 1-place

(3.1.6). If they have the core value adnv, they may be (a) elementary (fast) or phrasal (in the park), and (b) adnominal (tree in the park) or adverbial (walk in the park).[3]

3.2.2 Two Content kinds for the Semantic kind *PROPERTY*

concept	*indexical*	*name*
[sur: adj: great cat: adn ...]	[sur: adj: now cat: adnv ...]	
[sur: adj: enough cat: adnv ...]	[sur: adj: here cat: adnv ...]	
[sur: noun: table cat: snp ...]	[sur: noun: pro2 cat: sp2 ...]	
[sur: verb: melt cat: n-s3′ v ...]		

Of the Semantic kind *property*, only the concepts may be used nonliterally.[4]

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

3.2.3 One Content kind for 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.

[3] 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 or intransitive snore, which do not refer.

[4] For a nonliteral use of *on the table* see CC 9.2.4 and of *melt* CC 9.5.1.

3.3 Additional Constraint on Figurative Use

The restriction of figurative use to concepts is constrained further by the condition that the literal term and its figurative counterpart must be grammatically equivalent:

3.3.1 Invariance Constraint

> A figurative use and its literal counterpart must be of the same Syntactic and Semantic kind.

Thus, one cannot use a 1-place verb like bark to refer figuratively to a dog unless bark is nominalized, as in the little barker (i.e. by turning the property of intransitive bark into the referent barker, sleep into sleeper, stink into stinker, etc.). Similarly for the adj fat, which for figurative use must be nominalized, 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 in CC Chap. 9 all satisfy the invariance constraint:

3.3.2 Syntactic-semantic invariance of figurative use

Semantic kind	Syntactic kind	nonliteral use	literal counterpart	in CC
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

The Semantic kind *property* has several Syntactic kinds, while each Syntactic kind, e.g. prepnoun, has only one Semantic kind, i.e. *property*, regardless of whether it 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:

3.3.3 Example using all three Semantic kinds figuratively

> The animal flew acrobatically towards the disc.

The content obeys the invariance constraint: 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.

3.4 Declarative Specification of Concepts for Recognition

Concepts are the only Semantic kind which interacts directly with the agent's cognition-external environment. The interaction consists of matching between (i) raw data provided by sensors and activators of the agent's interface component and (ii) concept types provided by the agent's memory.[5]

Consider the rule for the recognition of a color:

3.4.1 Declarative specification for recognizing the color blue

	concept type		*concept token*
	place holder: blue		place holder: blue
	sensory modality: vision		sensory modality: vision
	semantic field: color		semantic field: color
	content kind: concept	⇒	content kind: concept
	wavelength: 450–495nm		wavelength: 470nm
	frequency: 670–610 THz		frequency: 637 THz
	samples: a, b, c, ...		samples: ...
	⇑		*analyzed output*
	sensory modality: vision		
raw input	sensor values: 470nm		
	637 THz		

The raw input data 470nm and 637 THz are provided by the agent's interface component and recognized 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 measurements of the input.

The place holder value of the recognized token, i.e. the letter sequence b l u e, is used for lookup of the lexical proplet which contains the place holder as its core value (CC 1.6.3):

[5] From a theory of science point of view, recognition and action based on concept types constitute an interaction between the humanities and the natural and engineering sciences.

3.4.2 PLACE HOLDER VALUE OF CONCEPT USED FOR LEXICAL LOOKUP

$$\mathsf{blue} \longrightarrow \begin{bmatrix} \text{sur:} \\ \text{adj: blue} \\ \text{cat: adnv} \\ \text{sem: pos} \\ \text{mdd:} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn:} \end{bmatrix}$$

Like the concept type, the proplet is retrieved from the artificial agent's memory (on-board database). Computationally, the lookup is based on string search (Knuth et al. 1977) in combination with a trie structure (Briandais 1959).

The language counterpart to the recognition of nonlanguage concepts is the interpretation of language-dependent surfaces. As an example, consider the DBS robot's recognition of the letter sequence blue by matching raw visual input data with letter patterns as shape types, resulting in a surface token:

3.4.3 SENSOR INTERACTING WITH LANG. CONCEPT TYPE (SURFACE)

$$\begin{array}{r} \text{sensory modality: vision} \\ \text{input: raw data} \\ \Downarrow \end{array}$$

$$\begin{array}{c} \textit{pattern: surface type} \\ \begin{bmatrix} \text{place holder: blue} \\ \text{sensory modality: vision} \\ \text{semantic field: language surface} \\ \text{content kind: roman letters} \\ \text{shape types: } \mathsf{b\ l\ u\ e} \\ \text{samples: ...} \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \textit{output: surface token} \\ \begin{bmatrix} \text{place holder: blue} \\ \text{sensory modality.: vision} \\ \text{semantic field: language surface} \\ \text{content kind: roman letters} \\ \text{shape tokens: } \mathsf{b\%\ l\%\ u\%\ e\%} \\ \text{samples: ...} \end{bmatrix} \end{array}$$

The input consists of raw data which are matched by the shape types of the letters.[6] The output replaces the shape types, here b l u e matching the raw data with the shape tokens, here b% l% u% e%, 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 sensory medium of speech. The shape types are used (i) for matching the raw data and (ii) for look-up of the lexical definition. For developing the linguistic side of automatic word form recognition, the type-token matching of raw data in different media may be cut short temporarily by typing letters directly into a standard computer.

[6] For ease of illustration, the letter shapes are represented by the letters themselves, e.g. e (type) and e% (token).

3.5 Declarative Specification of Concepts for Action

The action counterpart to the recognition of nonlanguage concepts is their external realization as raw data. It consists in adapting a type to the agent's purpose as a token which is passed to the appropriate actuator. As an example, consider a cuttlefish (Metasepia pfefferi) turning on the color blue:

3.5.1 Rule for producing the color blue

concept type

[place holder: blue
sensory modality: visual display
semantic field: color
content kind: concept
wavelength: 450–495nm
frequency: 670–610 THz
samples: a, b, c, ...]

⇒

concept token

[place holder: blue
sensory modality: visual display
semantic field: color
content kind: concept
wavelength: 470nm
frequency: 637 THz
samples: ...]

⇓

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 agent-selected values of 470nm and 637 THz. In the cuttlefish, these values are realized by natural actuators for color control (chromatophores) as raw data.

The language counterpart to a nonlanguage action is the realization of a language-dependent surface in a medium of choice. As an example, consider the DBS robot's production of the surface blue as raw data in vision:[7]

3.5.2 Realizing letter tokens as raw data in vision medium

surface type

[place holder: blue
sensory modality: vision
semantic field: language surface
content kind: roman letters
shape types: b l u e
samples: ...]

⇒

surface token

[place holder: blue
sensory modality.: vision
semantic field: language surface
content kind: roman letters
shape tokens: b% l% u% e%
samples: ...]

⇓

sensory modality: vision
actuator values: raw data *output*

The input to the actuator consists of a sequence of shape tokens representing roman letters. The output replaces the shape tokens, here b% l% u% e%, with matching raw data, for example on a computer screen.

[7] For ease of illustration, the letter shapes are represented by the letters themselves, e.g. e (type) and e% (token).

3.6 Indirect Grounding of Indexicals and Names

In DBS, the second computational Mechanism of *indexicals* is pointing at STAR values of the agent's on-board orientation system and is as such cognition-internal. However, because the STAR values originate as concept recognitions, past or present (CC Chaps. 7, 8), indexicals rely indirectly on the Mechanism of computational pattern matching. More specifically, the indexical pro1 points at the A value of the STAR, pro2 at the R value, pro3 at 3rd value, here at the S value, and now at the T value (with S and T values nominalized).

The third computational Mechanism of *names* relies on an act of baptism, which inserts the 'named referent' as the core value into a lexical name proplet (CTGR). Because the named referent originates as concept recognition, names – like indexicals – rely indirectly on the first computational Mechanism of *concepts*, i.e. computational pattern matching.

After working out the basic functioning of the computational Mechanism for the recognition and action of certain concepts in a codesigned but real environment, more concepts of the same kind (semantic field) may be added routinely, as shown by the following example:[8]

3.6.1 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'', ...

Once the recognition and action side of these concepts is working as intended, more colors may be easily added as an efficient, transparent upscaling.

Similarly for geometric forms: once the concepts of square (CC 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 (CC 2.5.1, 2), the robot should be able to execute language-based requests like Pick the blue square or Pick the green rectangle correctly from a set of items in its task environment.

Conclusion

A crucial requirement for data-driven agent-based DBS is *grounding*, i.e. the automatic interaction between the digital artificial cognition and the raw data

of the environment. In the absence of a robotics laboratory, recognition and action based on type-token matching between concepts and raw data in different media may be temporarily replaced by the shortcut of tapping input directly into a standard computer's key board and displaying output on the screen.

Thereby an artificial cognition may be developed as a declarative specification which is both, (i) readable by humans and (ii) allowing a straightforward translation into a general purpose programming language like Lisp, C, Java, or Perl. In computer science, a declarative specification represents the necessary properties of a software solution by omitting the accidental properties which make the individual programming languages different from each other. The software counterpart is the procedural implementation, serving as automatic verification.

Bibliography

Briandais, R. de la (1959) "File searching using variable length keys," *Proc. Western Joint Computer Conf. 15,* 295–298

CC = Hausser, R. (2019) *Computational Cognition; Integrated DBS Software Design for Data-Driven Cognitive Processing*, `lagrammar.net`

CTGR = Hausser, R. (2017a) "A computational treatment of generalized reference," *Complex Adaptive Systems Modeling*, Vol. 5.1:1–26.

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

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

Ogden, C.K., and I.A. Richards (1923) *The Meaning of Meaning*, London: Routledge and Kegan Paul

TExer3 = Hausser, R. (2020) *Twentyfour Exercises in Linguistic Analysis; DBS Software Design for the Hear and the Speak mode of a Talking Robot*, pp. 318. DOI: 10.13140/RG.2.2.13035.39200, `lagrammar.net`

[8] 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.

4. Comparing Coordination and Gapping in DBS

June 2, 2020

Abstract

The most basic distinction in the classical semantic relations of natural language is between (i) functor-argument and (ii) coordination. Functor-argument connects different kinds of contents, namely subject/predicate, object\predicate, and modifier|modified, while coordination connects the same kinds of content,[1] namely nominal−nominal, predicate−predicate, and modifier−modifier, at the elementary, phrasal, and in the case of predicates the clausal level of grammatical complexity (Sects. 4.1–4.6). Semantically related but different are the subject, predicate, and object gapping constructions, which are intrapropositional (Sects. 4.7–4.9).

Each example is systematically shown as (i) a content and (ii) a graphical representation of the semantic relations of structure. This brief but concise manner of analysis brings out the syntactic-semantic difference between coordination and gapping in general as well as the differences within the coordination examples and within the gapping examples in particular.

4.1 Coordination of Elementary Adnominals

In DBS, the distinction between functor-argument and coordination originates in the data structure of proplets, defined as non-recursive feature structures with ordered attributes. Core attributes of functor-argument are noun, verb, and adj. Attributes of coordination are nc (next conjunct) and pc (previous conjunct).

An example of a modifier−modifier coordination at the elementary level of grammatical complexity is tall, cool, black, new in the following content:

4.1.1 Content of The tall, cool, black, new building collapsed.

sur:	sur:	sur:	sur:	sur:	sur:
noun: building	adj: tall	adj: cool	adj: black	adj: new	verb: collapsed
cat: snp	cat: adn	cat: adn	cat: adn	cat: adn	cat: #n′ decl
sem: def sg	sem: pad	sem: pad	sem: pad	sem: pad	sem: ind past
fnc: collapse	mdd: building	mdd:	mdd:	mdd:	arg: building
mdr: tall	mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc: cool	nc: black	nc: new	nc:	nc:
pc:	pc:	pc: tall	pc: cool	pc: black	pc:
prn: 23	prn: 23	prn: 23	prn: 23	prn: 23	prn: 23

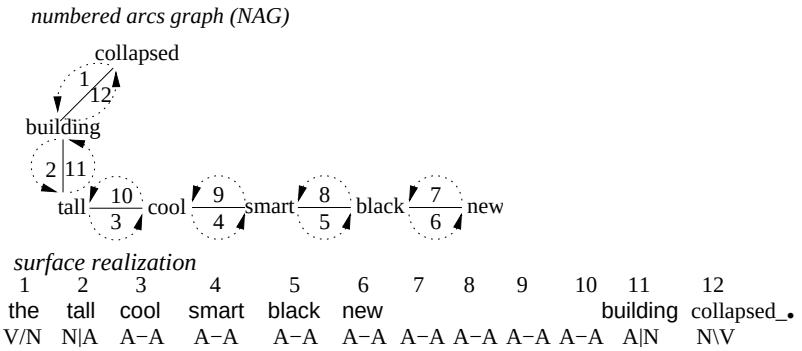
This content is a set (order-free) of completely self-contained proplets connected by (i) a shared prn value, here 23, and (ii) continuation values defined by address. The proplets of the adn coordination are connected and ordered by the values of their nc and pc attributes.

[1] For an overview of exceptions to the grammatical equality of conjuncts and proposals for their resolution see Bruening and Al Khalaf (2020).

The modification relation between the adn coordination *tall cool black new* and the noun *building* is tall|building. It is coded by the features [mdr: tall] of *building* and [mdd: building] of the initial conjunct *tall*. In the noninitial conjuncts, in contrast, the mdd attributes have no value; if needed, it can be retrieved from the initial conjunct via the pc connections (NLC Chap. 8).

The semantic relations coded in the content 4.1.1 may be shown as the following graph, whereby the different slashes /, |, and – represent the subject/predicate, modifier|modified and conjunct–conjunct relations:[2]

4.1.2 Graphical Analysis of the Semantic Relations



As shown in the *surface realization*, The is realized from the goal proplet in arc 1, tall from the goal proplet in arc 2, cool in arc 3, smart in arc 4, and new in arc 5. After empty return via the arcs 7-10, building is realized from the goal proplet of arc 11 and collapsed_. of arc 12. The navigation operations are named after the semantic relations they traverse and are shown in the bottom line, directly under the surface. The direction of the traversals is specified by the arc numbers directly above the surfaces.

4.2 Coordination of Phrasal Adnominal Modifiers

English phrasal modifiers consist of a preposition and a noun, e.g. in the water (noun concept), in Paris (noun name), or in here (noun indexical). In contrast to elementary modifiers, which may distinguish morphologically between adnominal and adverbial use, as in beautiful woman vs. sang beautifully, no such distinction exists in phrasal modifiers. Thus, in the water may be used adnominally (4.3.1) and adverbially (4.2.1). Also, while elementary modifiers in adnominal use precede the modified noun, phrasal modifiers follow:

[2] In linguistics, there is some agreement that the flat concatenation of coordination is a difficulty for substitution-based Phrase Structure Grammar (PSG): Ross (1967), Dik (1968), Goldsmith (1985), Sag, Gazdar, Wasow, and Weisler (1985), Lakoff (1986), Bayer (1996), Osborne (2006), and others.

4.2.1 Content of an adnominal modifier conjunction

The man in the water for days without a life jacket survived.

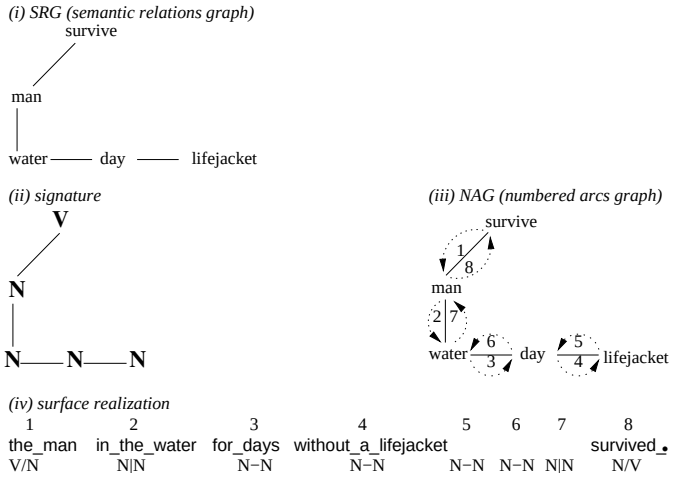
sur:	sur:	sur:	sur:	sur:
noun: man	noun: water	noun: day	noun: life jacket	verb: survive
cat: snp	cat: adnv	cat: adnv	cat: adnv	cat: #n′ decl
sem: def sg	sem: *in* def sg	sem: *for* indef pl	sem: *without* indef sg	sem: ind past
fnc: survive	fnc: survive	fnc:	mdd:	arg: man
mdr: water	mdd: man	mdr:	mdr:	mdr:
nc:	nc: day	nc: life jacket	nc:	nc:
pc:	pc:	pc: water	pc: day	pc:
prn: 26	prn: 26	prn: 26	prn: 26	prn: 26

By time-linear absorption of the determiner into the preposition and the noun into the preposition-determiner combination (CLaTR 7.2.5), a prepnoun is represented as a single proplet, like a case-marked locative in classical Latin. The core attribute is noun, the core value refers, the semantic role is specified by the cat value adnv, for adjective with adnominal and adverbial use, and the preposition is specified as the initial sem value of a prepnoun.

Prepnouns may be used (i) as phrasal (a) conjuncts and (b) modifiers, and (ii) in (a) adnominal and (b) adverbial use. The different uses are distinguished by word order in conjuncts (4.3.1 vs. 4.2.1), but create an ambiguity between an adnominal (TExer 1.5.3) and an adverbial (TExer 1.5.4) modifier reading.

They also differ in that the repetition of phrasal modifiers requires the same kind, whereas no such restriction holds for the repetition of phrasal conjuncts. For example, the modifier repetition on the table (locational) under the tree (locational) in the garden (locational) are all of the same kind (TExer Sect. 5.1), but the conjunct repetition in the water (locational) for days (temporal) without a life jacket (manner) are different.

4.2.2 Graph of an adnominal modifier conjunction



As in 4.1.2, the modifier|modified relation between the phrasal modifier coordination and the modified noun is traversed in arcs 2 (downward) and 7 (upward). Fulfillment of the continuity condition (NLC 3.6.5) as the think-speak mode counterpart to (and the source of) the time-linear derivation order in the hear mode is clearly shown in the bottom line of the *(iv) surface realization*, i.e. the goal proplet of operation n equals the start proplet of operation n+1.

4.3 Coordination of Phrasal Adverbial Modifiers

The distinction between the adnominal and the adverbial use of one and the same phrasal modifier coordination is located in the connection between the modified and the initial conjunct, e.g. between man and in the water in 4.2.1 (adnominal), and between survived and in the water in 4.2.2 (adverbial).

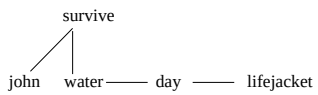
4.3.1 Content with an adverbial modifier conjunction

John survived in the water for days without a life jacket.

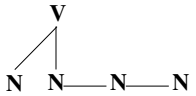
sur: john	sur:	sur:	sur:	sur:
noun: [person x]	verb: survive	noun: water	noun: days	noun: life jacket
cat: snp	cat: #n′ v	cat: snp	cat: snp	cat: snp
sem: def sg	sem: ind past	sem: *in* def sg	sem: *for* def sg	sem: *without* indef sg
fnc: survive	arg: [person x]	fnc: survive	mdd:	mdd:
mdr:	mdr: water	mdr:	mdr:	mdr:
nc:	nc:	nc: days	nc: life jacket	nc:
pc:	pc:	pc:	pc: water	pc: day
prn: 25	prn: 25	prn: 25	prn: 25	prn: 25

4.3.2 Graphical Analysis of the Semantic Relations

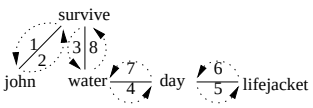
(i) SRG (semantic relations graph)



(ii) signature



(iii) NAG (numbered arcs graph)



(iv) surface realization

1	2	3	4	5	6	7	8	9
john	survived	in_the_water	for_days	without_a_lifejacket				•
V/N	N/V	V|N	N−N	N−N	N−N	N−N	N−N	N|V

The modifier|modified relation between the phrasal modifier coordination and the modified verb is traversed in arcs 3 (downward) and 8 (upward). Comparison with TExer 5.1.12 shows the semantic difference between the intrapropositional repetition of modification vs. coordination.

4.4 Coordination of Elementary Nouns as Subject

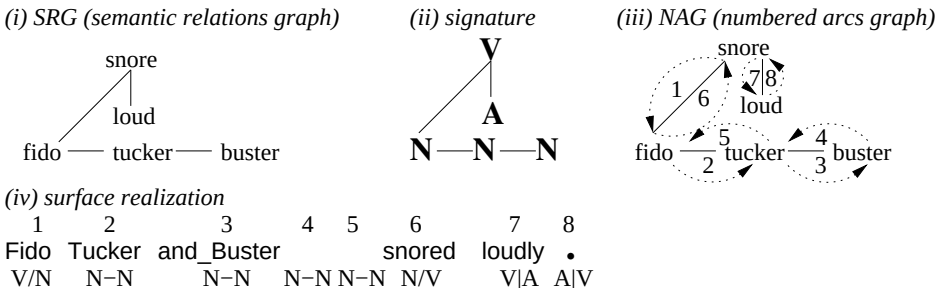
From elementary and phrasal adnominal coordination in Sects. 4.1–4.2, we turn to elementary noun coordination. The following coordination of names is coded via the nc and pc values and used as the grammatical subject:

4.4.1 Content of Fido, Tucker, and Buster snored loudly.

sur: fido	sur: tucker	sur: buster	sur:	sur:
noun: [dog x]	noun: [dog y]	noun: [dog z]	verb: snore	adj: loud
cat: snp	cat: snp	cat: snp	cat: #n′ decl	cat: adv
sem: nm m	sem: nm m	sem: *and* nm m	sem: ind past	sem: pad
fnc: snore	fnc:	fnc:	arg: [dog x]	mdd: snore
mdr:	mdr:	mdr:	mdr: loud	mdr:
nc: [dog y]	nc: [dog z]	nc:	nc:	nc:
pc:	pc: [dog x]	pc: [dog y]	pc:	pc:
prn: 18	prn: 18	prn: 18	prn: 18	prn: 18

In contrast to elementary and phrasal adnominal coordinations, the prefinal conjunction *and* in the coordination of nouns is obligatory.[3]

4.4.2 Graphical Analysis of the Semantic Relations



In a noun coordination, the connection to the functor-argument structure is based on the subject/predicate or the object\predicate relation, here arcs 1 and 6. In the content 4.4.1, this relation is coded by the [fnc: snore] feature of the initial conjunct *fido* and the [arg: [dog x]] feature of the verb *snore*. In the noninitial conjuncts, the fnc attributes have no value; if needed, it can be retrieved from the initial conjunct via the pc connections.

4.5 Intra- and Extrapropositional Verb Coordination

While adn and noun coordinations are always intrapropositional, verb coordination may be intra- or extrapropositional. Extrapropositional verb coordination is used for the concatenation of propositions, as in a text. The top verb of a

[3] For the graph analysis and for the complete sequence of explicit hear mode operations see TExer, Sect. 3.6.

possibly complex proposition represents the whole content because its continuation values allow a complete reconstruction. For example, the V↙/N traversal from *snore* to *fido* provides the initial conjunct of the subject coordination, the N→N traversal from *fido* to *tucker* provides the next conjunct, etc.[4]

Intra- and extrapropositional verb coordinations may combine as follows:

4.5.1 Intra- in extrapropositional verb coordination

Julia slept. Bob bought, peeled, and ate an apple. Fido snored.

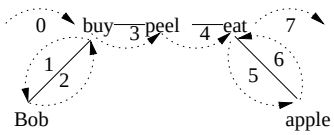
[prn: n] [prn: n+1] [prn: n+2]

The critical transition is from the intrapropositional coordination of proposition [prn: n+1] to the next proposition [prn: n+2] by means of an extrapropositional verb−verb coordination. The following solutions have been proposed:

4.5.2 Alternative NAG proposals for verb coordination

TExer3 proposal

(iii) numbered arcs graph (NAG)

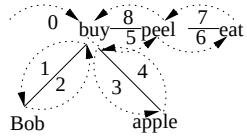


(iv) surface realization

1	2	3	4	5	6
Bob	bought	peeled	and_ate	an_apple	•
V/N^{i}	N /V	V−V	V−V	V\N	N\V

NLC2 proposal

(iii) numbered arcs graph (NAG)



(iv) surface realization

1	2	5	6	7	8	3	4
Bob	bought	peeled	and_ate			an_apple	•
V/N	N/V	V−V	V−V	V−V	V−V	V\N	N\V

The NLC2 analysis on the right takes an intrapropositional perspective by treating subject (Sect. 4.4), predicate (NLC2 8.3.4), object (NLC2 8.2.7), and modifier (Sects. 4.1–4.2) coordinations alike. Also, the initial conjunct *buy* is (a) the representative of the proposition as the carrier of the syntactic mood value, (b) the point of extrapropositional entrance, and (c) the point of extrapropositional exit.

If there is only a single top verb, which is usually the case, this is easily fulfilled. However, if there are several verbs of equal rank, as in the intrapropositional verb coordination of 4.5.1, there is a problem. A first symptom is the absent exit arc in the NLC2 NAG in 4.5.2. At the proplet level, the reason may be located in the nc slot of the initial verb proplet *buy*. Consider the following schematic instantiation:

[4] *Computational Cognition*, Sect. 3.3 Resonating Content: Selective Activation.

4.5.3 Intrapropositional verb-verb coordination

$$\begin{bmatrix}\text{verb: buy}\\ \ldots\\ \text{nc: peel}\\ \text{pc:}\\ \text{prn: n}\end{bmatrix}\begin{bmatrix}\text{verb: peel}\\ \ldots\\ \text{nc: eat}\\ \text{pc: buy}\\ \text{prn: n}\end{bmatrix}\begin{bmatrix}\text{verb: eat}\\ \ldots\\ \text{nc:}\\ \text{pc: peel}\\ \text{prn: n}\end{bmatrix}$$

The intrapropositional nature is shown by the shared prn value and the intrapropositional address values of the nc and pc attributes.

The nc slot of the first conjunct *buy* is filled by the address peel of the next intrapropositional verbal conjunct, which makes the slot unavailable for a relation to a next extrapropositional verbal conjunct (as would be needed for an NLC analysis of 4.5.1). The nc slot of the last conjunct, however, is available for an extrapropositional coordination with a next proposition.

Next consider the extrapropositional coordination 4.5.1 in a similar format as 4.5.3:

4.5.4 Extrapropositional verb-verb coordination

$$\begin{bmatrix}\text{verb: sleep}\\ \text{arg: julia}\\ \text{nc: (sing n+1)}\\ \text{pc:}\\ \text{prn: n}\end{bmatrix}\begin{bmatrix}\text{verb: buy}\\ \text{arg: bob apple}\\ \text{nc: (snore n+2)}\\ \text{pc: (sleep n)}\\ \text{prn: n+1}\end{bmatrix}\begin{bmatrix}\text{verb: snore}\\ \text{arg: fido}\\ \text{nc:}\\ \text{pc: (buy n+1)}\\ \text{prn: n+2}\end{bmatrix}$$

The extrapropositional nature of the coordination is shown by the extrapropositional address values of the nc and pc attributes, and the incremented prn values of the clausal conjuncts. Here the empty pc slot in the *sleep* proplet indicates initial position, just as an empty nc slot indicates the end of a text.

It is the empty nc slot of the intrapropositional coordination 4.5.3 which opens the possibility of the TExer3 analysis on the left of 4.5.2. The first verb of the intrapropositional coordination is the entrance proplet with (1) the subject value bob, (2) the object value apple and (3) the syntactic mood value decl, while the last verb *eat* of the coordination is the exit proplet. The non-initial intrapropositional conjuncts, here *peel* and *eat*, implicitly share the arg values and explicitly the prn value of the initial verbal conjunct *buy*. The only role *buy* had to relinquish is as the exit proplet, taken instead by the last verbal conjunct *eat* and expressed by the feature [nc: n+2] of *snore* (4.5.1).

The principle of using the last verbal proplet for the extrapropositional exit is not an exception, but applies to single top verb and multiple top verb propositions alike. It is just that in constructions with a single top verb the first verb proplet is also the last. Intrapropositionally, last verb proplets are clearly identified by their empty nc attribute.

The intrapropositional verb coordination in the TExer3 proposal on the left of 4.5.2 and the extrapropositional coordination in 4.5.1 have in common that they are uni-directional, in the direction of time. Anti-temporal traversal may be provided, however, by the following inference (shown here in the extrapropositional variant):

4.5.5 INFERENCE NAVIGATING BACKWARD THROUGH A COORDINATION

$$\begin{bmatrix} \text{verb: } \beta \\ \text{pc: } \alpha \\ \text{prn: n+1} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{verb: } \alpha \\ \text{nc: } \beta \\ \text{prn: n} \end{bmatrix}$$

While the functor-argument relations subject/predicate, object\predicate, and modifier|modified, as well as the coordination relations noun−noun and adj−adj are implemented from the outset for bidirectional traversal because it is always required, the V−V backward navigation by inference is used only when needed, as when telling a story starting from the end.

4.6 Extrapropositional Coordination

Extrapropositional coordination is needed for navigating from one sentence to the next, as in the two minimal propositions in the following minimal text:

4.6.1 CONTENT OF Mary slept. Fido snored.

$$\begin{bmatrix} \text{sur: mary} \\ \text{noun: [person x]} \\ \text{cat: snp} \\ \text{sem: nm f} \\ \text{fnc: sleep} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 17} \end{bmatrix} \begin{bmatrix} \text{sur:} \\ \text{verb: sleep} \\ \text{cat: \#ns3 decl} \\ \text{sem: ind past} \\ \text{arg: [person x]} \\ \text{mdr:} \\ \text{nc: (snore 18)} \\ \text{pc:} \\ \text{prn: 17} \end{bmatrix} \begin{bmatrix} \text{sur:} \\ \text{verb: snore} \\ \text{cat: \#ns3 decl} \\ \text{sem: ind past} \\ \text{arg: [dog y]} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc: (sleep 17)} \\ \text{prn: 18} \end{bmatrix} \begin{bmatrix} \text{sur: fido} \\ \text{noun: [dog y]} \\ \text{cat: snp} \\ \text{sem: nm m} \\ \text{fnc: snore} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 18} \end{bmatrix}$$

The two propositions are distinguished by their prn values 17 and 18.

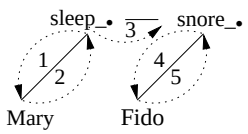
All proplets have nc (next conjunct) and pc (previous conjunct) attributes which may be empty or take an intra- or extrapropositional address value. Here, the nc attribute of *sleep* has the extrapropositional address value (snore 18) while the pc attribute of *snore* has the extrapropositional address value (sleep 17).

The pivot of the hear mode derivation is the interpunctuation between the

two propositions. As shown in TExer3 Sect. 2.1, the interpunctuation proplet (i) supplies the syntactic mood value to the top verb of the present sentence, (ii) cross-copies with the intervening subject of the next sentence, and (iii) absorbs the next verb and thus becomes the predicate of the next sentence. These steps leave no trace in the content 4.6.1.

4.6.2 Graphical Analysis of the Semantic Relations

(iii) numbered arcs graph (NAG)



(iv) surface realization

1	2	3	4	5
Mary	slept_•		Fido	snored_•
V/N^{i}	N/V	V−V	V/N	N/V

This completes our discussion of coordination in the narrow sense.

4.7 Quasi Coordination in Subject Gapping

In linguistics, a grammatical construction in which a single *shared* item is in a semantic relation with a sequence of n (1≤n) *gapped* items (repetition) is called *gapping*. Basic examples are (i) subject gapping, (ii) predicate gapping, and (iii) object gapping, which have the following pretheoretical structure:

4.7.1 Pretheoretical comparison of three gapping kinds

subject gapping	*predicate gapping*	*object gapping*
Bob buy apple	Bob **buy** apple	Bob buy ∅
∅ peel pear	Jim ∅ pear	Jim peel ∅
and ∅ eat peach	*and* Bill ∅ peach	*and* Bill eat **peach**

The shared item is shown in bold face, while the gapped items are indicated by the gap marker ∅.[5]

The following example shows the content of a subject gapping:

4.7.2 **Bob** bought an apple, ∅ peeled a pear, and ∅ ate a peach.

```
[sur: bob                  ]  [sur:                    ]  [sur:                  ]  [sur:                    ]
[noun: [person x]          ]  [verb: buy               ]  [noun: apple           ]  [verb: peel              ]
[cat: snp                  ]  [cat: #n′ #a′ decl       ]  [cat: snp              ]  [cat: #n′ #a′ v          ]
[sem: nm m                 ]  [sem: ind past           ]  [sem: indef sg         ]  [sem: ind past           ]
[fnc: buy                  ]  [arg: [person x] apple   ]  [fnc: buy              ]  [arg: [person x] pear    ]
[      peel                ]  [mdr:                    ]  [mdr:                  ]  [mdr:                    ]
[      eat                 ]  [nc:                     ]  [nc:                   ]  [nc:                     ]
[mdr:                      ]  [pc:                     ]  [pc:                   ]  [pc:                     ]
[nc:                       ]  [prn: 32                 ]  [prn: 32               ]  [prn: 32                 ]
[pc:                       ]
[prn: 32                   ]
```

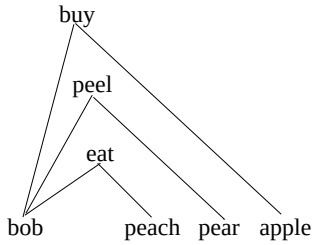
sur:	sur:	sur:
noun: pear	verb: eat	noun: peach
cat: snp	cat: #n′ #a′ v	cat: snp
sem: indef sg	sem: *and* ind past	sem: indef sg
fnc: peel	arg: [person x] peach	fnc: eat
mdr:	mdr:	mdr:
nc:	nc:	nc:
pc:	pc:	pc:
prn: 32	prn: 32	prn: 32

Gapping constructions are only quasi coordinations because the nc and pc slots do not have intrapropositional values. They resemble nominal and intrapropositional verb coordinations, however, in that they use prefinal *and* and consist of unbounded repetitions of grammatically similar items.[6]

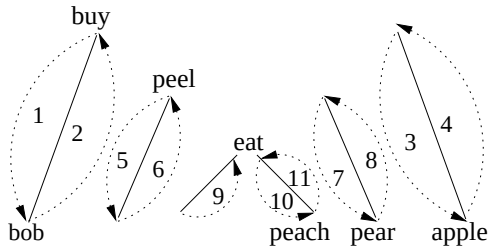
The semantic relations between the shared item *bob* and the gapped items ∅ *peel pair* and ∅ *eat peach* are run via the gap list in the shared item and the repetition of the shared item's address, here [person x], in the verbs of the gapped items (arg slot, initial position). In this way, the semantic relations of structure are complete even in a gapping construction.[7]

4.7.3 Graphical Analysis of the Semantic Relations

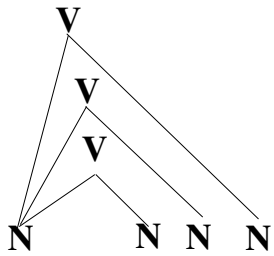
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	3	4	1	6	7	8	5	9	10	11
Bob	bought	an_apple			peeled	a_pear			and_ate	a_peach	.
V/N^{i}	N^{s}/V	V\N	N\V	V/N^{s}	N^{s}/V	V\N	N\V	V/N^{s}	$N^{s}\backslash V$	V\N	N\V

[5] There seems to be no "modifier gapping" in natural language.

The different tilts of the three N/V and N\V relations are solely for visual separation in the graph. The gaps appear as the empty traversals of the arcs 4, 1, and 8, 5. The navigation ends with arc 11. The upward arc 9 does not have a downward counterpart. The arc numbering is breadth-first. The number of operations is even. As a multiple verb construction (Sect. 4.5), the last verb, here *eat*, is used for the extrapropositional exit.

The think-speak navigation along the semantic relations between proplets is continuous (Continuity Condition, NLC 3.6.5), as shown by the bottom line of the *surface realization*. This is possible by leaving the control of the gaps in the surface, here arcs 4, 1 and 8, 5, to the lexicalization rules. For example, lexnoun realizes the surface of the shared noun proplet *bob* (goal proplet of the $V\swarrow N^s$ operations in arcs 1, 1, and 5) if, and only if, its initial fnc value is not yet #-marked.

4.8 Quasi Coordination in Predicate Gapping

The pretheoretical characterization of predicate gapping in 4.7.1 is formally instantiated as the following content:

4.8.1 Bob **bought** an apple, Jim ∅ a pear, and Bill ∅ a peach**.**

sur: bob	sur:	sur:	sur: jim	sur:
noun: [p. x]	verb: buy	noun: apple	noun: [p. y]	noun: pear
cat: snp	cat: #n′ #a′ decl	cat: snp	cat: snp	cat: snp
sem: nm m	sem: ind past	sem: indef sg	sem: nm m	sem: indef sg
fnc: buy	arg: [p. x] apple	fnc: buy	fnc: buy	fnc: buy
mdr:	[p. y] pear	mdr:	mdr:	mdr:
nc:	[p. z] peach	nc:	nc:	nc:
pc:	...	pc:	pc:	pc:
prn: 33	prn: 33	prn: 33	prn: 33	prn: 33

sur: bill	sur:
noun: [p. z]	noun: peach
cat: snp	cat: snp
sem: *and* nm m	sem: indef sg
fnc: buy	fnc: buy
mdr:	mdr:
nc:	nc:
pc:	pc:
prn: 33	prn: 33

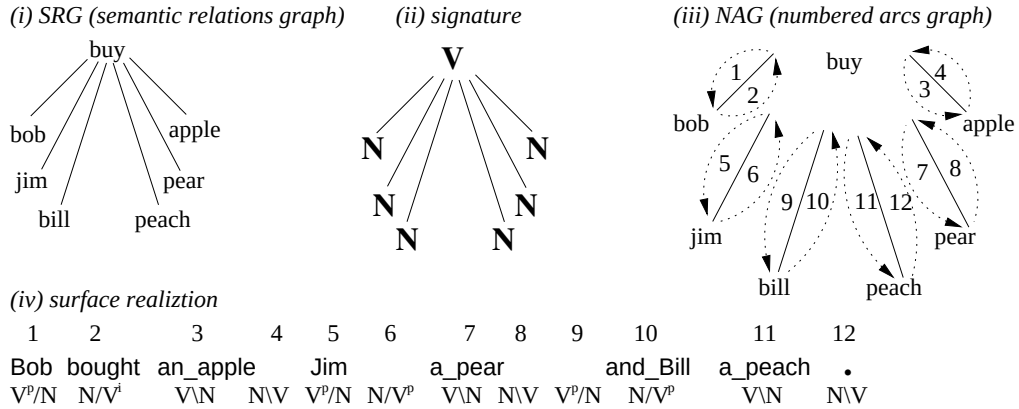
The shared item is *buy*. Its arg slot contains the gap list, here the subject-object pairs *bob apple, jim pear,* and *bill peach*. The conjunction *and* is coded into

[6] In linguistics, there is some agreement that gapping does not correspond to constituent structure: Ross (1967, 1970), Jackendoff (1971), Kuno (1976), Sag (1976), Hankamer (1979), McCawley (1988), Hartmann (2000), Osborne (2006), Johnson (2009), and others.

[7] For the detailed DBS analysis see TExer3, Sect. 5.2.

the initial sem slot of *bill*. The subject and object proplets of the gapped items take buy as their shared fnc value.

4.8.2 Graphical Analysis of the Semantic Relations



The shared predicate relates to the subject and object of its initial sentence (arcs 1–4) and of its two gapped items (arcs 5–8 and 9–12).

4.9 Quasi Coordination in Object Gapping

Compared to subject and predicate gapping, in which the gaps precede the shared item (filler), object gapping is special in that the filler follows the gaps. Therefore the gap list must be accumulated in an external cache until the filler arrives.

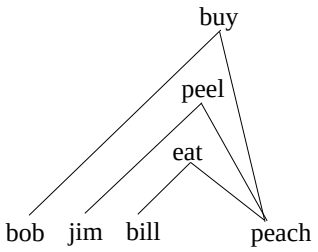
4.9.1 Bob bought ∅, Jim peeled ∅, and Bill ate **a peach .**

```
[sur: bob          ]  [sur:                      ]  [sur: jim          ]  [sur:                      ]
[noun: [person x]  ]  [verb: buy                 ]  [noun: [person y]  ]  [verb: peel                 ]
[cat: snp          ]  [cat: #n′ #a′ v            ]  [cat: snp          ]  [cat: #n′ #a′ v            ]
[sem: nm m         ]  [sem: ind past             ]  [sem: nm m         ]  [sem: ind past             ]
[fnc: buy          ]  [arg: [person x] peach     ]  [fnc: peel         ]  [arg: [person y] peach     ]
[...               ]  [...                       ]  [...               ]  [...                       ]
[prn: 34           ]  [prn: 34                   ]  [prn: 34           ]  [prn: 34                   ]

[sur: bill         ]  [sur:                      ]  [sur:              ]
[noun: [person z]  ]  [verb: eat                 ]  [noun: peach       ]
[cat: snp          ]  [cat: #n′ #a′ decl         ]  [cat: snp          ]
[sem: nm m         ]  [sem: and ind past         ]  [sem: indef sg     ]
[fnc: eat          ]  [arg: [person z] peach     ]  [fnc: [p. x] buy   ]
[mdr:              ]  [mdr:                      ]        [p. y] peel  ]
[nc:               ]  [nc:                       ]        [p. z] eat   ]
[pc:               ]  [pc:                       ]  [...               ]
[prn: 34           ]  [prn: 34                   ]  [prn: 34           ]
```

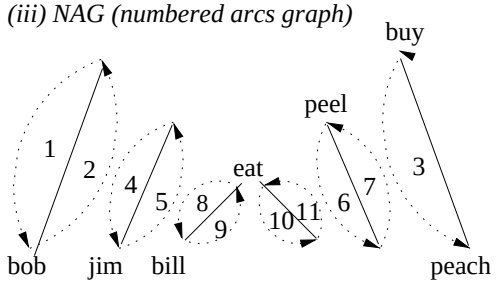
The three verb proplets all take the core value peach of the shared object as their non-initial arg value.

4.9.2 Graphical Analysis of the Semantic Relations

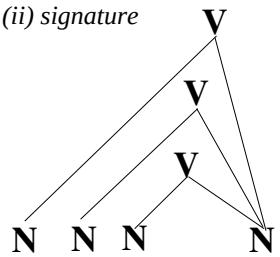
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	3	7	4	5	6	11	8	9	10	11
Bob	bought			Jim	peeled			and_Bill	ate	a_peach	•
V/N	N/V	$V\backslash N^o$	$N^o\backslash V$	V/N	N/V	$V\backslash N^o$	$N^o\backslash V$	V/N	N/V	$V\backslash N^f$	$N^f\backslash V$

The shared object is clearly shown. Just as the graph 4.7.3 for subject gapping is missing a downward arc opposite arc 9, the current graph for object gapping is missing an upward arc opposite arc 3. As a multiverb construction, the last verb, here *eat*, is used for the extrapropositional exit.

4.10 Conclusion

Coordination and gapping have in common that they repeat an unlimited number of similar items. They differ in that the connection between the conjuncts of a coordination is coded by the values of their nc (next conjunct) and pc (previous conjunct) attributes, while no such nc−pc relations exist in gapping constructions.

Instead subject, predicate, and object gapping establish the relation between a single *shared item* and a sequence of repeating *gapped items* by means of (i) a gap list in the shared item and (ii) copies of the shared item's core value in the grammatically appropriate slots of the gapped items. The nc and pc attributes are not used, i.e. they have no intrapropositional values in gapping constructions.

Bibliography

Bayer, S. (1996) "The coordination of unlike categories," *Language*, Vol. 72:579–616

Bruening, B., and E. Al Khalaf (2020) "Category mismatches in coordination revisited," *Linguistic Inquiry*, Vol. 31.1:1–36

Culicover, P. and R. Jackendoff (1997) "Semantic subordination despite syntactic coordination," *Linguistic Inquiry*, Vol. 28.2:195–217

Dik, S. (1968) *Coordination: Its Implications for a Theory of General Linguistics*, Amsterdam: North Holland Publishing Company

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

Gazdar, G., I. Sag, T. Wasow & S. Weisler (1985) "Coordination and how to distinguish categories," in *Natural Language and Linguistic Theory 3*

Goldsmith, J. (1985) "A principled exception of the coordinate structure constraint," CLS 21

Hankamer, J. (1979) *Deletion in Coordinate Structures*, New York: Garland Publishing

Hartmann, K. (2000) *Right Node Raising and Gapping: Interface Conditions on Prosodic Deletion*, Amsterdam: John Benjamins

Jackendoff, R. (1971) "Gapping and related rules," *Linguistic Inquiry*, Vol. 2: 21–35

Johnson, K. (2009) "Gapping is not (VP) ellipsis," *Linguistic Inquiry*, Vol. 40:289–328

Kuno, S. (1976) "Gapping: A functional analysis," *Linguistic Inquiry*, Vol. 7:300–318

Lakoff, G. (1986) "Frame semantic control of the coordinate structure constraint," CLS 22

McCawley, J. (1988) *The syntactic phenomena of English,* Chicago: The University of Chicago Press

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

Osborne, T. (2006) "Gapping vs. non-gapping coordination," *Linguistische Berichte*, Vol. 207:307–338

Ross, J.R. (1967) *Constraints on Variables in Syntax*, Doctoral Dissertation, MIT (published as 'Infinite syntax!' Ablex, Norwood (1986))

Ross, J.R. (1970)"Gapping and the order of constituents," in M. Bierwisch and K. Heidolph (eds.), *Progress in linguistics: A collection of papers*, pp. 249–259, The Hague: Mouton

Sag, I. (1976) *Deletion and Logical Form*, Doctoral Dissertation, MIT, Cambridge, Massachusetts

TExer3 = Hausser, R. (2020) *Twentyfour Exercises in Linguistic Analysis, DBS software design for the Hear and the Speak mode of a Talking Robot*, pp. 318. `lagrammar.net`

5. Are Iterating Slot-Filler Structures Universal?

Dec 8, 2019

Abstract

An example of iterating slot-fillers[1] is repeating infinitives as in John decided to try to persuade Mary to sing (Sect. 5.3). Each infinitive serves as a phrasal object with a slot, e.g. to try *slot*. The slot is filled by another infinitive with a slot, i.e. to persuade Mary *slot*, and so on, except for the last one, which terminates the iteration by means of an intransitive verb. Similar slot-filler repetitions are iterating object clauses (Sect. 5.4) and iterating adnominal clauses (Sect. 5.5).

Another kind of iterating slot-filler or filler-slot structure is known as gapping. For example, Bill bought an apple, peeled a pear, ..., and ate a peach. is called subject gapping. Here the open-ended sequence *slot* peeled a pear, *slot* ate a peach, an so on, takes Bill as the shared subject. The end of the iteration is announced by the function word and in penultimate position. Systematic variants are predicate gapping and object gapping (Sect. 5.6).

Iterating slot-filler structures may interact with other grammatical constructions. A prime example is a *long distance dependency*, such as Whom did John say that Bill claims that Suzy believes that ... Mary loves?, i.e. the single filler-slot pair Whom and Mary loves? is separated by a slot-filler iteration of object clauses (Sect. 5.7).

Remarkably, all of these highly conspicuous constructions occur in natural languages which are completely unrelated to English and other European languages, namely in Korean, Tagalog, and Georgian. They are therefore candidates for being universal thought structures.

keywords: hear mode, content, speak mode, iteration, marked vs. unmarked slots, infinitive, object clause, adnominal clause, gapping, long distance dependency, intra- vs. extrapropositional relations

5.1 Language and Thought

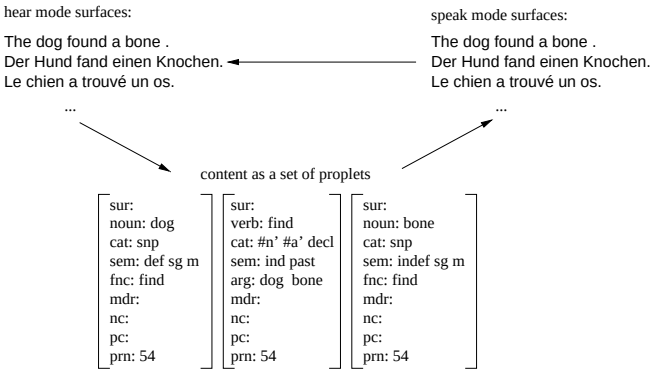
DBS is based on a data-driven agent-based ontology[2] which treats cognitive contents as sets (order-free) of proplets, defined as nonrecursive feature structures with ordered attributes. Proplets are connected into contents by the classical semantic relations of structure coded by address. Language content and thought content are treated alike except that the proplets of a language content have language-dependent sur(face) values which are absent in the proplets of a thought content. A given content may use different surfaces from typologically similar languages, for example, The dog found a bone, Der Hund fand einen Knochen, and Le chien a trouvé un os.

The following graph shows one and the same content as input to the speak mode and as output of the hear mode for different languages:

[1] McCord (1980) used the slot-filler principle for *Slot Grammar*.

[2] The alternative is the substitution-driven sign-based ontology, used by Symbolic Logic and Generative Grammar.

5.1.1 From content to speak mode to hear mode to content

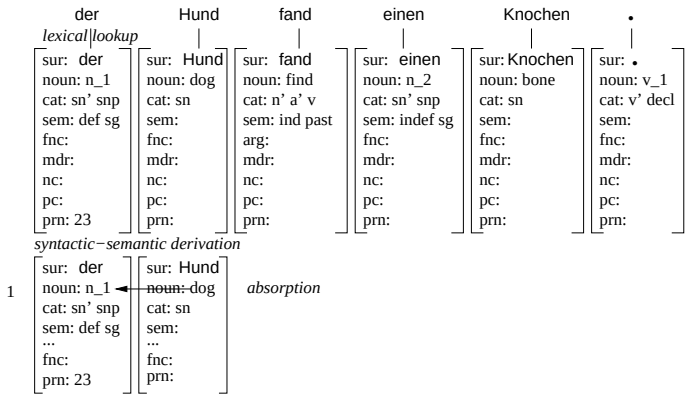


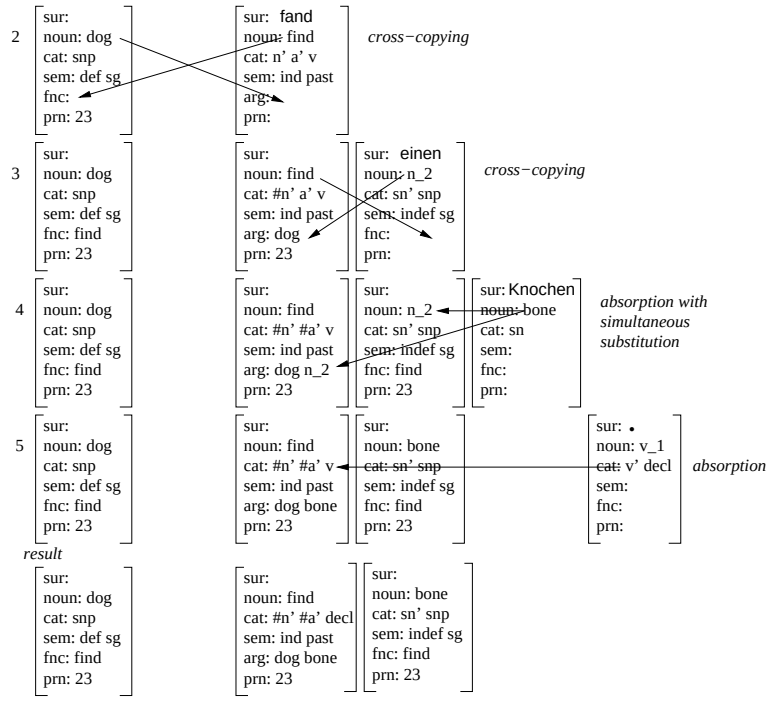
In language communication, automatic word form recognition provides lexical proplets. They are connected by the hear mode into content, which may be used as input to inferencing or the speak mode for surface production.

As the computational data structure of DBS, a proplet encodes all lexical and compositional properties of a content word as proplet-internal attribute-value pairs called features. For example, the noun proplets in 5.1.1 have the lexical features [sem: def sg] and [sem: indef sg] for definiteness and number, and the verb proplet has the lexical features [cat: #n′ #a′ decl] for valency and syntactic mood, and [sem: past ind] for tense and verbal mood. The syntactic-semantic connections are coded inside the verb proplet by the value order in the feature [arg: dog bone]. The subject proplet *dog* and the object proplet *bone* are connected to the *find* proplet by their respective [fnc: find] features.

Consider the explicit hear mode derivation of the German example in 5.1.1:

5.1.2 Surface compositional time-linear hear mode derivation

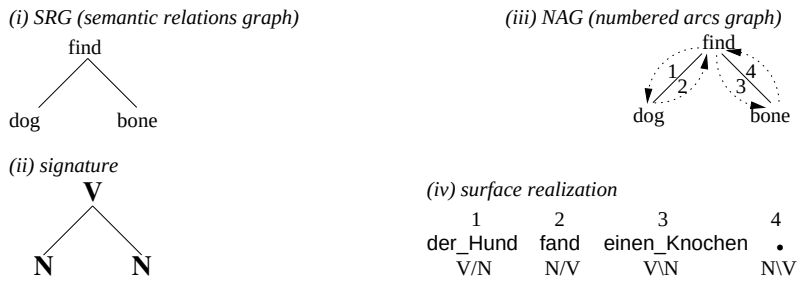




The derivation is *surface compositional* because each input surface has exactly one lexical proplet representation and there are no lexical proplet representations without a concrete surface. The derivation is *time-linear* as shown by the stair-like addition of one new next word form in each line. The proplets of the function words the and a absorb their respective content words, as shown in lines (1,2) and (4,5), while the . proplet is absorbed into the top verb (5, result).

Contents resulting from hear mode derivations are possible inputs to the think mode operations of (a) selective activation by navigation and (b) inferencing. Either may be mirrored by language-dependent surfaces in the speak mode riding piggyback on sequences of think mode operations:

5.1.3 Semantic relations underlying speak mode derivation

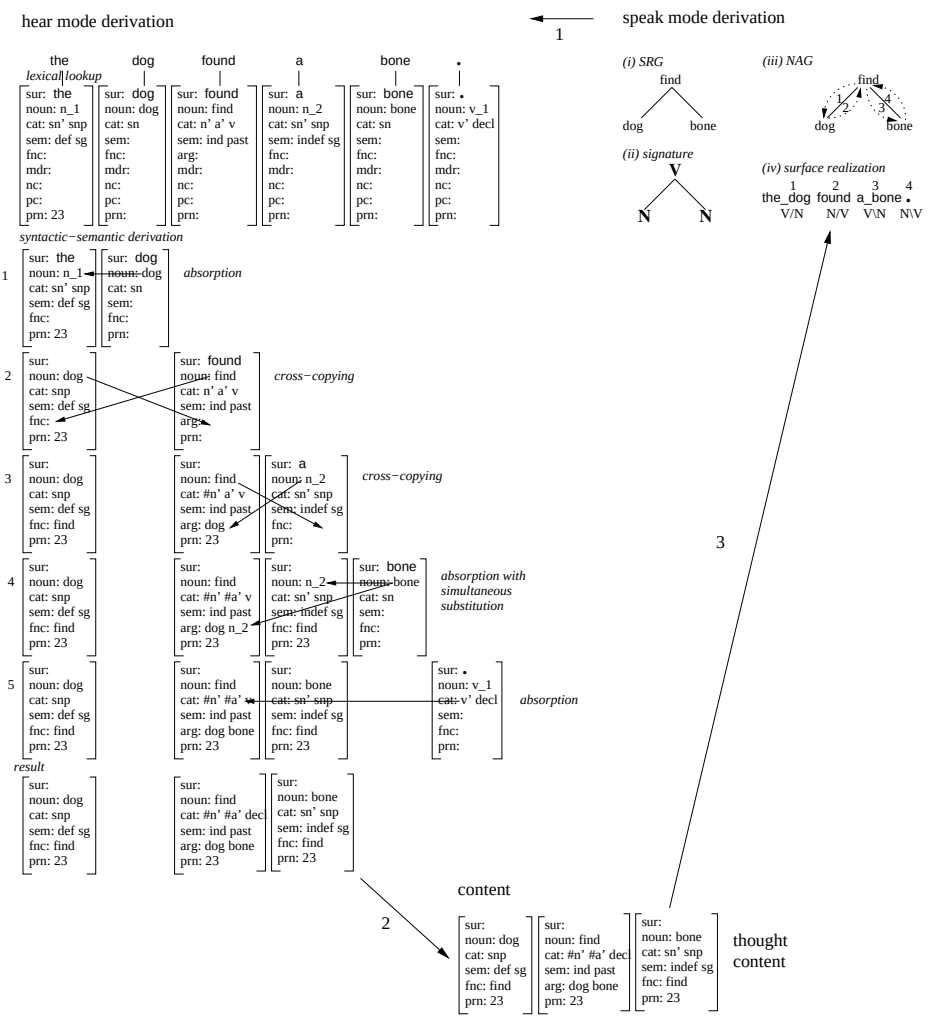


The *(i) SRG* and the *(ii) signature* show the static semantic structure, here subject/predicate and object\predicate, whereby the nodes in the *SRG* rep-

resent the core values and in the *signature* the core attributes of the proplets in the content. The *(iii) NAG* and the *(iv) surface realization*, in contrast, show the dynamic aspect of the think mode which activates content by a navigation for inferencing and for speak mode realization of language-dependent surfaces.

The hear mode derivation 5.1.2, the content 5.1.1, and the speak mode derivation 5.1.3 combine into the following cycle of natural language communication in DBS (shown for English surfaces):

5.1.4 Cycle of natural language communication



This constellation supports two important functions of turn taking (Schegloff 2007): (i) transfer of a content from the speaker to the hearer (two agents, arrow sequence 3, 1) and (ii) interpreting content in the hear mode and reproducing it in the speak mode (one agent, arrow sequence 2, 3).

5.2 Slot-Filler Iteration

There is comparatively little argument that, roughly speaking, all natural languages distinguish between the word kinds noun, verb, and adj, the syntactic moods declarative, interrogative, and imperative, the verbal moods indicative and subjunctive, the tenses present, past, and future, the semantic relations of functor-argument and coordination, the degrees of elementary, phrasal, and clausal grammatical complexity, and the intra- vs. extrapropositional semantic relations of structure, whereby the latter may be intra- or extrasentential. The basic grammatical structures built on these notions may be considered likely to be universal contents.

In contrast, structures perhaps more promising for being nonuniversal is the conspicuous iteration of slot-fillers or filler-slots[3] such as the following:

5.2.1 Examples of different slot-filler iterations

1. *Iterating infinitives* (phrasal)
   John decided to try to persuade Mary to sing.
2. *Iterating object clauses* (clausal)
   Mary saw that Peter saw that Suzy saw Fido.
3. *Iterating adnominal clauses* (clausal)
   Mary saw the man who loves the woman who feeds the child.
4. *Subject gapping* (phrasal)
   Bob bought an apple, peeled a pear, and ate a peach.
5. *Predicate gapping* (phrasal)
   Bob bought an apple, Jim a pear, and Bill a peach.
6. *Object gapping* (phrasal)
   Bob bought, Jim peeled, and Bill ate the peach.
7. *Object clause iteration with long distance dependency* (clausal)
   Whom did John say that Bill believes that Mary claims that Suzy loves?

In this list, three construction kinds may be distinguished: (i) marked slot-filler repetition using different fillers (1-3), (ii) unmarked filler-slot or slot-filler repetition using the same filler (4-6), and (iii) a single filler-slot relation with an intervening object clause iteration, resulting in a long-distance dependency.

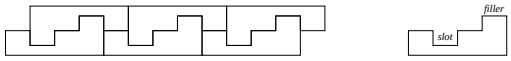
5.3 Marked Slot-Filler Repetition in Infinitives

In English, marked slot-filler repetition occurs at the phrasal level as (a) repeating infinitives,[4] and at the clausal level as (b) repeating object clauses and (c) repeating adnominal clauses. The cognitive structure common to all three may be shown abstractly as follows:

[3] Perhaps because they are of low frequency and not in every tree bank.

[4] In HPSG, infinitives are treated as a kind of clause (Sag 1997). It seems, however, that the adnominal use of infinitives, as in The decision to try..., is restricted to nominalized transitive verbs.

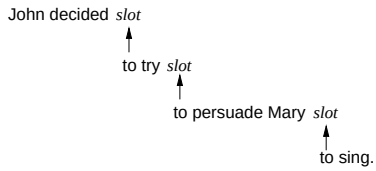
5.3.1 SLOT-FILLER REPETITION AS ABSTRACT COGNITIVE STRUCTURE



For example, the slot-fillers in an infinitive iteration consist of a sequence of transitive predicates taking another transitive predicate as their second argument, except for the last one. The lower predicate's subject is absent in the surface (slot) but equals implicitly the next higher subject (subject control, e.g. try, promise) or the next higher object (object control, e.g. persuade).

The function word marking the slots in English is to. The characteristic semantic connections in iterating infinitives may be illustrated as follows:

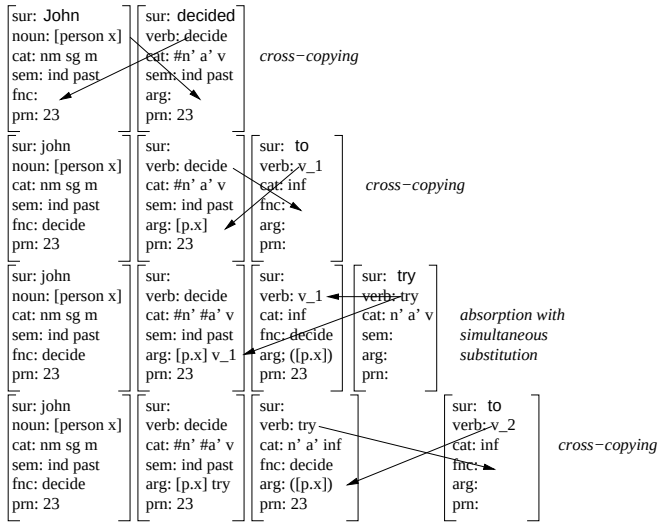
5.3.2 MARKED SLOTS IN ITERATING INFINITIVES

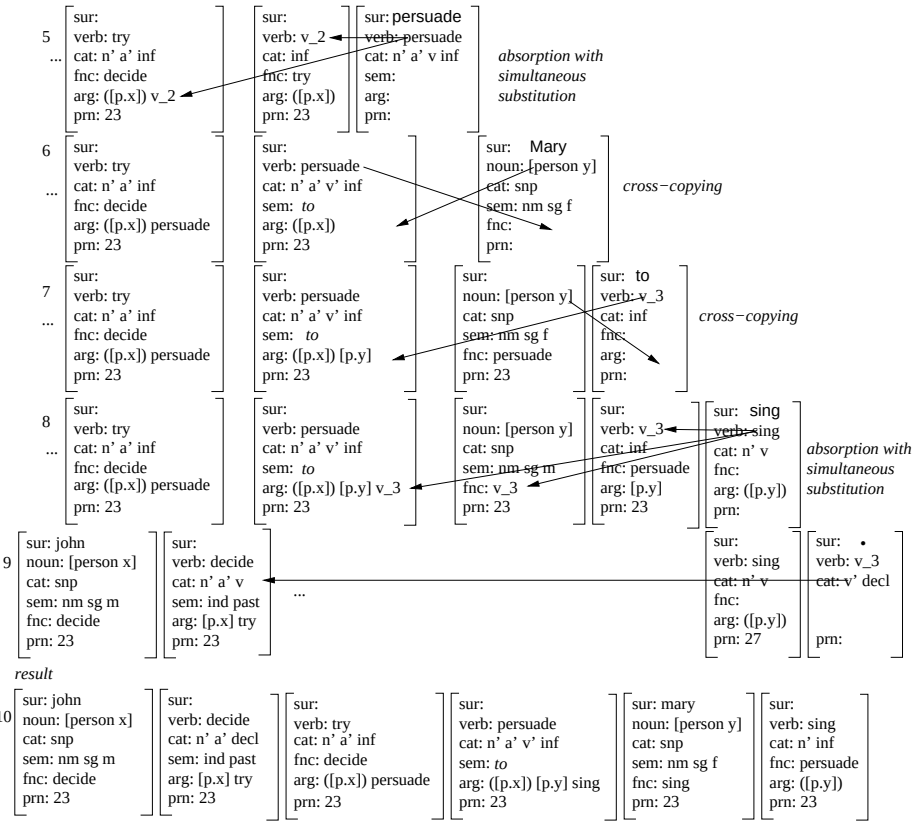


The initial predicate decide_*slot* is filled by to try_*slot* which is filled by to persuade Mary_*slot* which is filled by to sing. The last filler terminates the iteration because it does not introduce another slot. Whether an infinitive has subject or object control (CLaTR Sects. 15.4–15.6) depends on the verb.

Consider the surface compositional time-linear derivation of John decided to try to persuade Mary to sing.:

5.3.3 HEAR MODE DERIVATION OF REPEATING INFINITIVES



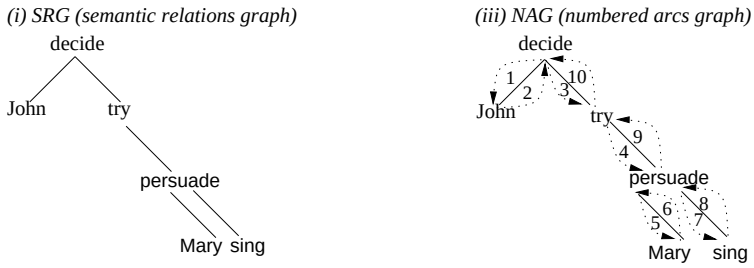


As a surface compositional approach, DBS connects decide directly with to_try, try directly with to_persuade, and persuade directly with to_sing, using the intrapropositional V\V relation. The implicit subject and object control of the infinitives is shown explicitly as the values [person x] and [person y], i.e. the named referents of John and Mary (in round brackets).

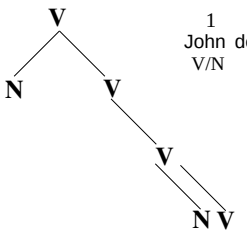
For the corresponding speak mode, consider the following graph analysis:

5.3.4 Syntactic-Semantic Analysis of Infinitive Repetition

John decided to try to persuade Mary to sing.



(ii) signature *(iv) surface realization*



1	2	3	4	5	6	7	8	9	10
John	decided	to_try	to_persuade	Mary		to_sing			.
V/N	N/V	V\V	V\V	V\N	N/V	V\V	V\V	V\V	V\V

content

$$\begin{bmatrix}\text{sur: john}\\ \text{noun: [person x]}\\ \text{cat: nm sg f}\\ \text{sem:}\\ \text{fnc: decide}\\ \ldots\\ \text{prn: 32}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: decide}\\ \text{cat: n}'\text{ a}'\text{ decl}\\ \text{sem: ind past}\\ \text{arg: [person x] try}\\ \ldots\\ \text{prn: 32}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: try}\\ \text{cat: n}'\text{ a}'\text{ inf}\\ \text{sem: } \textit{to}\\ \text{arg: ([person x]) persuade}\\ \ldots\\ \text{prn: 32}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: persuade}\\ \text{cat: n}'\text{ a}'\text{ v}'\text{ inf}\\ \text{sem: } \textit{to}\\ \text{arg: ([person x])[person y] sing}\\ \ldots\\ \text{prn:32}\end{bmatrix}$$

$$\begin{bmatrix}\text{sur: mary}\\ \text{noun: [person y]}\\ \text{cat: snp}\\ \text{sem: nm sg m}\\ \text{fnc: sing}\\ \ldots\\ \text{prn: 32}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: sing}\\ \text{cat: n}'\text{ a}'\text{ inf}\\ \text{sem: } \textit{to}\\ \text{arg: ([person y])}\\ \ldots\\ \text{prn: 32}\end{bmatrix}$$

The first two infinitives have subject control, while the third has object control. Of the verbs, sing is 1-place, decide and try are 2-place, and persuade is 3-place. The valency positions are marked with $'$. The #-marking is omitted.

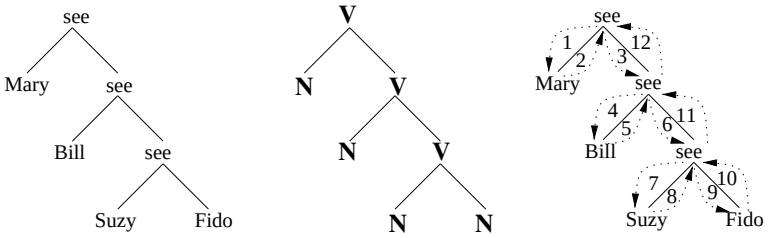
5.4 Marked Slot-Filler Repetition in Object Clauses

The clausal counterpart to phrasal infinitive iteration is the extrapropositional repetition of object sentences, as in the following example:

5.4.1 Object clause repetition

Mary saw that Bill saw that Suzy saw Fido.

(i) SRG (semantic relations graph) *(ii) signature* *(iii) NAG (numbered arcs graph)*



(iv) surface realization

1	2	3	4	5	6	7	8	9	10	11	12
Mary	saw	that	Bill	saw	that	Suzy	saw	Fido			.
V/N	N/V	V\V	V/N	N/V	V\V	V/N	N/V	V\N	N/V	V\V	V\V

content

sur: mary	sur:	sur: bill	sur:
noun: [person x]	verb: see	noun: [person y]	verb: see
cat: snp	cat: #n′ #a′ decl	cat: snp	cat: #ns3′ #a′ v
sem: nm sg f	sem: *that* ind past	sem: nm sg m	sem: *that* ind past
fnc: see	arg: [person x] (see 34)	fnc: see	arg: [person y] (see 35)
...	...	...	...
prn: 33	prn: 33	prn: 34	prn: 34

sur: suzy	sur:	sur: fido
noun: [person z]	verb: see	noun: [dog w]
cat: snp	cat: #ns3′ #a′ v	cat: snp
sem: nm sg m	sem: *that* ind past	sem: nm sg m
fnc: see	arg: [person z] (see 35)	fnc: see
...	...	...
prn: 35	prn: 35	prn: 35

The extrapropositional nature of the example is shown by the different prn values, from 33 to 35. The function word marking the slots is that:

5.4.2 Marked slots in iterating object clauses

Mary saw *slot*
↑
that Bill saw *slot*
↑
that Suzy saw Fido.

The last clause terminates the iteration because the filler does not introduce another slot. The analyses 5.3.3, 5.3.4, and 5.4.1 rely on the strictly time-linear derivation order of DBS.

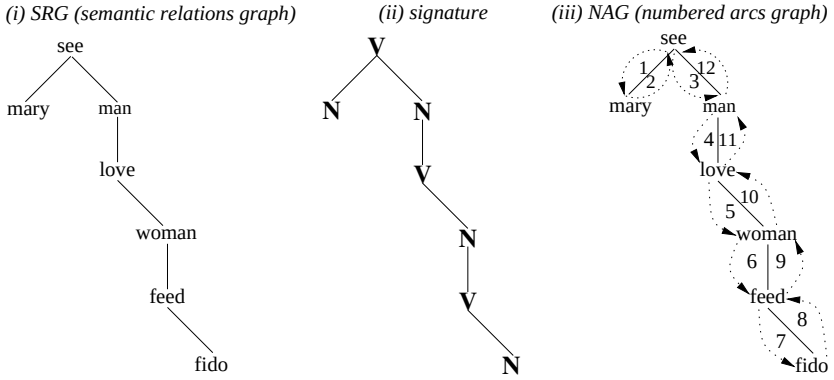
5.5 Marked Slot-Filler Repetition in Adnominal Clauses

Like infinitives and object clauses, adnominal clauses may be iterated, as in Mary saw the man who loves the woman who ... feeds Fido.

5.5.1 Graph analysis underlying multiple adnominal clauses

Mary saw the man who loves the woman who fed Fido.

(i) SRG (semantic relations graph) *(ii) signature* *(iii) NAG (numbered arcs graph)*



(iv) surface realization (English, subject gap)

1	2	3	4	5	6	7	8	9	10	11	12
Mary	saw	the_man	who_loves	the_woman	who_fed	Fido					•
V/N	N/V	V\N	N|V	V\N	N|V	V\N	N\V	V|N	N\V	V|N	N\V

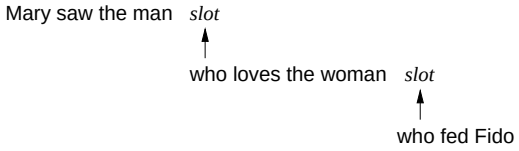
content

sur: mary	sur:	sur:	sur:	sur:
noun: [person x]	verb: see	noun: man	verb: love	noun: woman
cat: snp	cat: #n′ #a′ decl	cat: snp	cat: #ns3′ #a′ v	cat: snp
sem: nm f	sem: past	sem: def sg	sem: *who* pres	sem: def sg
fnc: see	arg: [person x] man	fnc: see	arg: ∅ woman	fnc: love
mdr:	mdr:	mdr: (love 30)	mdd: [man 29]	mdr: (feed 31)
prn: 29	prn: 29	prn: 29	prn: 30	prn: 30

sur:	sur: fido
verb: feed	noun: [dog x]
cat: #n′ #a v	cat: snp
sem: *who* past	sem: nm
arg: ∅ [dog x]	fnc: feed
mdd: [woman 30)	mdr:
prn: 31	prn: 31

In this construction, the function word marking the slot is a "relative pronoun" (subordinating conjunction with argument role) such as who, whom or which. As shown by the increasing prn values, the construction is extrapropositional.

5.5.2 Marked slot structure of iterating adnominal clauses



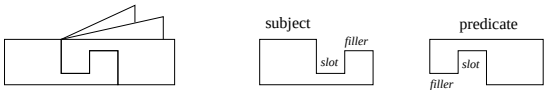
The function word marking the slots in this example is who.

5.6 Unmarked Slot-Filler Iteration in Gapping Constructions

In gapping constructions, a single unmarked slot is used several times, as in subject and predicate gapping (filler-slot) and object gapping (slot-filler). This is in contradistinction to slot-filler iterations which use several different marked slots (Sects. 5.3–5.5).

In subject gapping, a single subject slot takes multiple predicate fillers, which may be shown as follows:

5.6.1 Subject gapping as an abstract cognitive structure



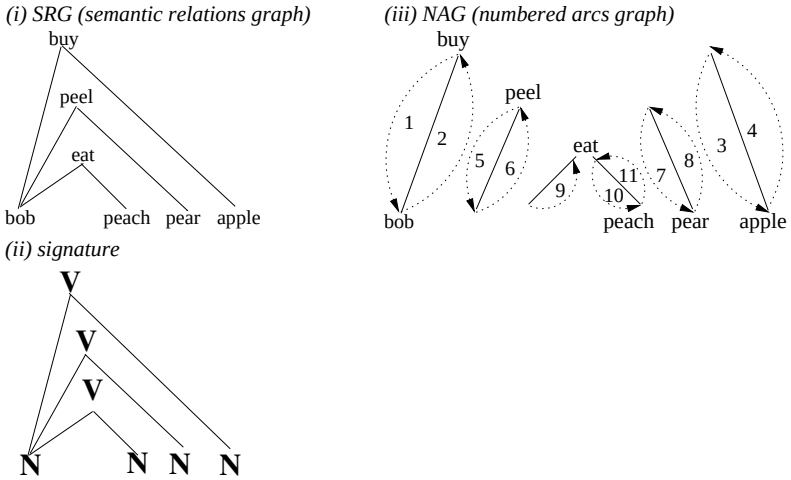
Without the multiple predicates, this three-dimensional graph would equal a simple subject-predicate combination, i.e. a subject with a single intransitive

or transitive verb. Computationally, the slot-filler combination is implemented as a cross-copying between order-free proplets.

Instead of marking the slots with a function word, as in slot-filler iteration, gapping marks the slots with a pause in speech or a comma in writing. This kind of slot is called ‘unmarked,’ but indicated by ∅ for analysis:

5.6.2 DBS ANALYSIS OF SUBJECT GAPPING (TExer Sect. 5.2)

Bob bought an apple ∅ peeled a pear, and ∅ ate a peach**.**



(iv) surface realization

1	2	3	4	1	6	7	8	5	9	10	11
Bob	bought	an_apple			peeled	a_pear			and_ate	a_peach	.
V/N^{i}	N^{s}/V	$V\backslash N$	$N\backslash V$	V/N^{s}	N^{s}/V	$V\backslash N$	$N\backslash V$	V/N^{s}	$N^{s}\backslash V$	$V\backslash N$	$N\backslash V$

content

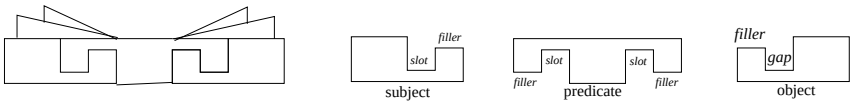
```
[sur: bob
 noun: [person x]
 cat: snp
 sem: nm m
 fnc: buy
      peel
      eat
 ...
 prn: 32]

[sur:
 verb: buy
 cat: #n′ #a′ decl
 sem: past
 arg: [person x] apple
 ...
 prn: 32]

[sur:
 noun: apple
 cat: snp
 sem: indef sg
 fnc: buy
 ...
 prn: 32]

[sur:
 verb: peel
 cat: #n′ #a′ v
 sem: past
 arg: [person x] pear
 ...
 prn: 32]

[sur:
 noun: pear
 cat: snp
 sem: indef sg
 fnc: peel
 ...
 prn: 32]

[sur:
 verb: eat
 cat: #n′ #a′ v
 sem: and past
 arg: [person x] peach
 ...
 prn: 32]

[sur:
 noun: peach
 cat: snp
 sem: indef sg
 fnc: eat
 ...
 prn: 32]
```

In contradistinction to the depth-first numbering of the NAGs in marked slot-filler constructions, the NAG numbering in gapping constructions is breadth-first (TExer Chap. 1). The final gapped item is announced by the function word *and*, coded in the sem slot of the *eat* proplet.

The proplet *bob* serves as the shared subject of the predicates buy, peel, and eat by specifying them in its fnc slot as the *gap list*. The inverse relation from the predicates to their shared subject is established by writing the core value [person x] of *bob* into the first (subject) arg slot of the verbs. The construction is treated as intrapropositional, as indicated by the common prn value 32.

The next gapping construction is predicate gapping, in which a single predicate takes multiple subjects and objects as fillers (three-dimensional):

5.6.3 Predicate gapping as an abstract thought structure

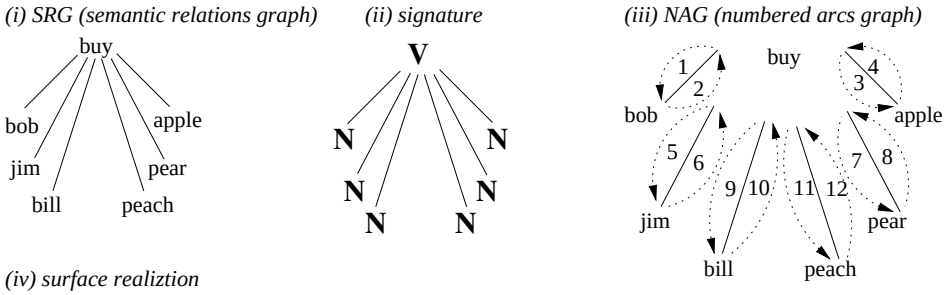


Without the multiple subjects and objects, the graph would equal a simple subject/predicate\object combination.

The canonical DBS graph analysis of predicate gapping may be shown as follows:

5.6.4 DBS analysis of predicate gapping (TExer Sect. 5.3)

Bob **bought** an apple, Jim ∅ a pear, and Bill ∅ a peach.



(iv) surface realiztion

1	2	3	4	5	6	7	8	9	10	11	12
Bob	bought	an_apple		Jim		a_pear			and_Bill	a_peach	.
V^p/N	N/V^i	V\N	N\V	V^p/N	N/V^p	V\N	N\V	V^p/N	N/V^p	V\N	N\V

content

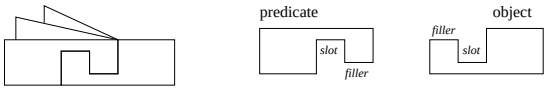
```
[sur: Bob
 noun: [person x]
 cat: snp
 sem: nm m
 fnc: buy
 ...
 prn: 33]

[sur:
 verb: buy
 cat: #n′ #a′ decl
 sem: past
 arg: [person x] apple
      [person y] pear
      [person z] peach
 ...
 prn: 33]

[sur:
 noun: apple
 cat: snp
 sem: indef sg
 fnc: buy
 ...
 prn: 33]

[sur: Jim
 noun: [person y]
 cat: snp
 sem: nm m
 fnc: buy
 ...
 prn: 33]

[sur:
 noun: pear
 cat: snp
 sem: indef sg
 fnc: buy
 dots
 prn: 33]
```

sur: Bill	sur:
noun: [person z]	noun: peach
cat: snp	cat: snp
sem: *and* nm m	sem: indef sg
fnc: buy	fnc: buy
dots	...
prn: 33	prn: 33

The shared item is the predicate *buy*. Its arg slot contains the gap list, here the subject-object pairs *bob apple, jim pear,* and *bill peach*. The conjunction *and* is coded into the initial sem slot of *bill*. The subject and object proplets take buy as their shared fnc value.

Our last gapping construction is object gapping in which a single object is taken by multiple predicates. Its graph is the mirror image of the three-dimensional graph for subject gapping (5.6.1):

5.6.5 Object gapping as an abstract thought structure

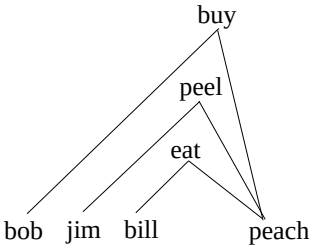


Without the multiple predicates, the graph would equal a simple object\predicate combination (including a predicate without a subject, as in imperatives).

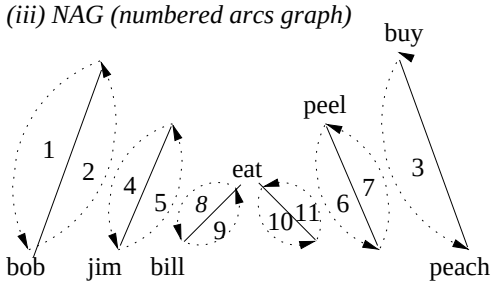
5.6.6 DBS analysis of object gapping (TExer Sect. 5.4)

Bob bought ∅, Jim peeled ∅, and Bill ate a peach

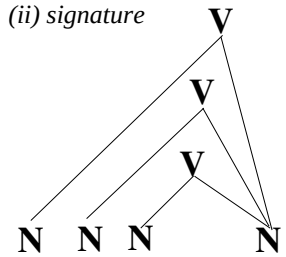
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

1	2	3	7	4	5	6	11	8	9	10	11
Bob	bought			Jim	peeled			and_Bill	ate	a_peach	•
V/N	N/V	V\N^o	N^o\V	V/N	N/V	V\N^o	N^o\V	V/N	N/V	V\N^f	N^f\V

content

sur: Bob	sur:	sur: Jim	sur:
noun: [person x]	verb: buy	noun: [person y]	verb: peel
cat: snp	cat: #n′ #a′ v	cat: snp	cat: #n′ #a′ v
sem: nm m	sem: past	sem: nm m	sem: past
fnc: buy	arg: [person x] peach	fnc: peel	arg: [person y] peach
…	…	…	…
prn: 34	prn: 34	prn: 34	prn: 34

sur: Bill	sur:	sur:
noun: [person z]	verb: eat	noun: peach
cat: snp	cat: #n′ #a′ decl	cat: snp
sem: *and* nm m	sem: past	sem: indef sg
fnc: eat	arg: [person z] peach	fnc: buy
…	…	peel
prn: 34	prn: 34	eat
		…
		prn: 34

Gapping constructions are intrapropositional, like iterated infinitives.

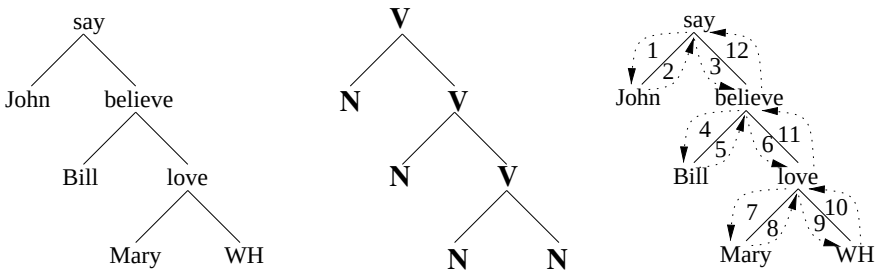
5.7 Long-Distance Dependency

The slot-filler iteration of object clauses (Sect. 5.4) may be combined with a single filler-slot relation, taking the filler in first and the slot in last position. This results in the following long-distance dependency (TExer Sect. 5.5):

5.7.1 Object-clause iteration with long-distance dependency

Whom did John say that Bill believes that Mary loves?

(i) SRG (semantic relations graph) *(ii) signature* *(iii) NAG (numbered arcs graph)*



(iv) surface realization

3	6	9	10	11	12	1	2	3	4	5	6	7	8	11	12
	Whom				does	John	say	that	Bill	believes	that	Mary	loves		?
V\V	V\V	V\N	N\V	V\V	V\V	V/N	N/V	V\V	V/N	N/V	V\V	V/N	N/V	V\V	V\V

content

sur:	sur:	sur: john	sur:
noun: wh_1	verb: say	noun: [person x]	verb: believe
cat: obq	cat: #ns3′ #a′ interrog	cat: snp	cat: #ns3′ #a′ v
sem: *whom*	sem: *do* pres	sem: nm m	sem: *that* pres
fnc: love	arg: [person x] (believe 28)	fnc: say	arg: [person y] (love 26&)
…	…	…	fnc: (say 27)
prn: 26&	prn: 27	prn: 27	…
			prn: 28

sur: bill	sur:	sur: mary
noun: [person y]	verb: love	noun: [person z]
cat: snp	cat: #ns3′ #a′ v	cat: snp
sem: nm m	sem: *that* pres	sem: nm f
fnc: believe	arg: [person z] wh_1	fnc: love
...	fnc: (believe 28)	...
prn: 28	...	prn: 26&
	prn: 26&	

As in the object clause iteration 5.4.1, the slots are marked with the function word that.

In DBS, an unbounded dependency has the form of an unbounded suspension, here between initial whom and the final transitive verb form loves. Because the length of the suspension is only determined at the end of the input sentence, the time-linear surface compositional hear mode derivation has the following syntactic ambiguity structure (NLC 7.6.5):

5.7.2 Ambiguity structure of an unbounded suspension

```
A  Whom   does  John   love?
    1 ——— 1 ——— 1 ——— 1
B         does  John   say    that   Bill   loves?
           2 ——— 2 ——— 2 ——— 1 ——— 1 ——— 1
C                        that   Bill   believes  that   Mary   loves?
                          3 ——— 3 ——— 3 ——— 1 ——— 1 ——— 1
D                                            that   Mary   claims  that  Suzy  loves?
                                              4 ——— 4 ——— 4 ——— 1 ——— 1 ——— 1
```

This systematic ambiguity does not affect the linear time complexity of DBS because the time-linear derivation replaces each previous reading with the new input. Accordingly, 5.7.2 has only a single reading, namely D.

Conclusion

This paper investigates grammatical structures which are conspicuous in that they are based on repeated sharing. The question is whether these constructions are (i) limited to a certain language type or (ii) universal in that they may be found in all natural languages. The latter hypothesis is supported by the fact that the iterations discussed have direct structural counterparts in several languages completely unrelated to English, namely Korean[5], Tagalog[6], and Georgian[7]. This is remarkable, and analyzed data from additional languages supporting or opposing the conjecture would be interesting.

[5] Thanks to Professor Kiyong Lee, native speaker of Korean.
[6] Thanks to Mrs. Guerly Söllch, M.A., native speaker of Tagalog.
[7] Thanks to Mrs. Sofia Tekmalaze-Fornwald, M.A., native speaker of Georgian.

Bibliography

CC = Hausser, R. (2019) *Computational Cognition; Integrated DBS Software Design for Data-Driven Cognitive Processing*, `lagrammar.net`

Delshad, F. (2009) *Georgica et Irano-Semitica, Philologische Studien zu den iranischen und semitischen Elementen im georgischen Nationalepos 'Der Recke im Pantherfell'*, Schriften zur Literaturwissenschaft, Wiesbaden `https://www.db-thueringen.de/servlets/MCRFileNodeServlet/dbt_derivate_00003610/Diss.pdf`

McCord, M.C. (1980) "Slot Grammars," *American Journal of Computational Linguistics*, Vol. 6.1:31–43

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

Sag, I. (1997) "English relative clause constructions," *Journal of Linguistics*, Vol. 33.2:431-48

Schegloff, E.A. (2007) *Sequence Organization in Interaction,* New York: Cambridge U. Press

TExer3 = Hausser, R. (2020) *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, `lagrammar.net`

6. Computational Pragmatics

Oct. 23, 2019

Abstract

In DBS, a content is built by connecting the three semantic kinds of natural cognition, namely concepts, indexicals, and names, with the classical semantic relations of structure, namely subject/predicate, object\predicate, modifier|modified, and conjunct−conjunct, at the elementary, phrasal, and clausal level of grammatical complexity. A content with a language-dependent surface for inter-agent communication is called a meaning$_1$ (literal meaning).

The first step of a computational[1] pragmatic interpretation is anchoring a content type to the agent's on-board orientation system, resulting in a content token.[2] The current state of the orientation system is coded as a nonrecursive feature structure called STAR, which monitors the agent's location S(pace), T(ime), A(gent's name), and R(ecipient, partner of discourse), plus 3rd (other), and prn (proposition number). In addition to monitoring, the STAR provides values for indexicals in the content to point at (Sects. 6.1, 6.2).

The second step is coactivating *resonating content* in the agent's on-board database (A-memory), using different degrees of token line intersections (Sect. 6.3). The third step is exploring the neighborhood of a resonating content by *selective activation* (Sect. 6.4). The remaining Sects. 6.5–6.7 address the pragmatics of literal and nonliteral use.

keywords: language vs. nonlanguage content, content types vs. content tokens, on-board orientation system (STAR), literal vs. nonliteral use, resonating content, token line intersection, selective activation

6.1 Four Kinds of Content in DBS

There are four kinds of content in DBS, characterized by the binary values ±surface and ±STAR. Nonlanguage contents are −surface and language contents are +surface. Content types are −STAR and content tokens +STAR.[3]. Consider the following examples, beginning with the nonlanguage type, i.e. [−surface, −STAR], of The dog found a bone.:

[1] For overviews of noncomputational pragmatics see Kempson (2001) and Horn&Ward eds. (2004).

[2] A type defines the *necessary* properties of a concept, while a token is an instantiation of the type with additional *accidental* properties (Peirce 1906, CP Vol.4, p. 375).

[3] The STAR in agent-based DBS may be seen as a development of the sign-based "parameter approach," which uses parameters such as I (possible worlds), J (possible times), S (possible speakers), H (possible hearers), etc., such as the index @, I, J, g superscripted to logical formulas in Montague (1973). Cresswell (1972, p.4) wonders tongue in cheek about adding a next drink parameter.

6.1.1 [−surface, −STAR]: NONLANGUAGE CONTENT TYPE

$$\begin{bmatrix}\text{sur:}\\ \text{noun: dog}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: find}\\ \text{cat: \#n}'\ \text{\#a}'\ \text{decl}\\ \text{sem: past ind}\\ \text{arg: dog bone}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{noun: bone}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}$$

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

The next example is the corresponding nonlanguage token:

6.1.2 [−surface, +STAR]: NONLANGUAGE CONTENT TOKEN

$$\begin{bmatrix}\text{sur:}\\ \text{noun: dog}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{verb: find}\\ \text{cat: \#n}'\ \text{\#a}'\ \text{decl}\\ \text{sem: past ind}\\ \text{arg: dog bone}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{sur:}\\ \text{noun: bone}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: 12}\end{bmatrix}\begin{bmatrix}\text{S: yard}\\ \text{T: friday}\\ \text{A: sylvester}\\ \text{R:}\\ \text{3rd:}\\ \text{prn: 12}\end{bmatrix}$$

The three content proplets and the STAR proplet are connected by a common prn constant, here 12. According to the STAR, the token resulted as an observation by the agent Sylvester on Friday in the yard.

The language content type corresponding to 6.1.1 illustrates the independence of language-dependent sur values, here German, from the relatively language-independent placeholders (English base forms for convenience):

6.1.3 [+surface, −STAR]: LANGUAGE CONTENT TYPE

$$\begin{bmatrix}\text{sur: der\_Hund}\\ \text{noun: dog}\\ \text{cat: snp}\\ \text{sem: def sg}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{sur: fand}\\ \text{verb: find}\\ \text{cat: \#n}'\ \text{\#a}'\ \text{decl}\\ \text{sem: past ind}\\ \text{arg: dog bone}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}\begin{bmatrix}\text{sur: einen\_Knochen}\\ \text{noun: bone}\\ \text{cat: snp}\\ \text{sem: indef sg}\\ \text{fnc: find}\\ \text{mdr:}\\ \text{nc:}\\ \text{pc:}\\ \text{prn: K}\end{bmatrix}$$

A language content type is also called a literal meaning$_1$. It is an abstraction in that an actual DBS hear mode derivation results in a content token.[4] The

[4] A content type may be systematically obtained from a content token by removing the STAR and replacing the prn constants with suitable variables.

content types 6.1.1 and 6.1.3 match an open-ended number of corresponding tokens with different prn values.

Finally consider a language content token which matches the type 6.1.3. Produced by the speaker sylvester in German towards the intended hearer tweety, it corresponds to the nonlanguage content token 6.1.2 except for the R value:

6.1.4 [+surface, +STAR]: LANGUAGE CONTENT TOKEN

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	sem: indef sg	R: tweety
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 surface sequence passed from speaker to hearer is a token of raw data. For communication to be successful, the hearer must reconstruct not only (i) the literal meaning$_1$ but also the (ii) speaker meaning$_2$ adjusted to the hearer's perspective, resulting in a STAR-2 content (Sect. 6.5).

6.2 Context of Interpretation Depends on Agent's Database

The second step of a computational pragmatic interpretation is the automatic coactivation of resonating content in the agent's on-board database. It relies on (a) the database schema of the agent's A-memory and (b) the data structure of proplets. The database schema is defined as follows:

6.2.1 TWO-DIMENSIONAL DATABASE SCHEMA OF A-MEMORY IN DBS

- *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.

The time-linear arrival order of 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.[5]

[5] 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 DBS A-memory, formerly called word bank.

6.2.2 TOKENLINE WITH CLEARED NOW FRONT

(i) member proplets		*(ii) now front*	*(iii) owner*
noun: bone … fnc: fetch … prn: 2	noun: bone … fnc: hide … prn: 5		bone

The result of Sylvester's assembling the observation 6.1.2 is as follows:

6.2.3 CONTENT 6.1.2 AT THE NOW FRONT OF SYLVESTER'S A-MEMORY

	(i) member proplets		*(ii) now front*	*(iii) owners*
…	sur: noun: bone cat: snp sem: indef sg fnc: fetch … prn: 2	sur: noun: bone cat: snp sem: def sg fnc: hide … prn: 5	… sur: noun: bone cat: snp sem: indef sg fnc: find … prn: 12	bone
	…			
…	sur: noun: dog cat: snp sem: indef sg fnc: bury … prn: 3	sur: noun: dog cat: pnp sem: def pl fnc: eat … prn: 7	… sur: noun: dog cat: snp sem: def sg fnc: find … prn: 12	dog
	…			
…	sur: verb: find cat: #n′ #a′ decl sem: pres prog arg: (person x) square … prn: 5	sur: verb: find cat: #n′ #a′ decl sem: past ind arg: cat mouse … prn: 8	… sur: verb: find cat: #n′ #a′ decl sem: past ind arg: dog bone … prn: 12	find
	…			
…	S: living room T: monday A: sylvester R: tweety 3rd: Speedy prn: 2	… S: garden T: thursday A: sylvester R: hector 3rd: prn: 10	… S: yard T: friday A: sylvester R: 3rd: prn: 12	sylvester

The proplets are order-free, but connected with semantic relations of structure coded by intrapropositional address.[6]

[6] The address of a proplet is defined as the core and prn value, e.g. (dog 12). Because the prn value of an intrapropositional address equals the prn value of the proplet in which it is contained, it is redundant and therefore omitted, as shown in 6.1.2 and 6.1.4. Extrapropositionally, however, the prn value of an address is not redundant and must be specified explicitly, as shown in 6.4.3.

When the proplets at the now front have ceased to be candidates for additional intrapropositional concatenations, the now front is cleared by moving it and the owners one step to the right into fresh memory territory (loom-like clearance). By leaving the proplets of the completed proposition behind (here with the prn value 12), they become member proplets by switching their location to permanent storage, never to be changed, like sediment, but freely available for reading and copying, and correction by adding, like a diary entry.

The A-memory illustrated in 6.2.3 is content-addressable for the following reasons. First, it does not use a separate index (catalog), unlike a coordinate-addressable database, e.g. an RDBMS. Second, the 'content' used for a proplet's storage in, and retrieval from, the agent's memory is the letter sequence of the proplet's core value (which enables computational string search in combination with a trie structure).

6.3 Coactivating Content by Token Line Intersection

Tokens at the now front may be turned automatically into search patterns for coactivating resonating content in A-memory. The technical basis for this procedure is the database schema of A-memory with its token lines, now front, and owners (6.2.2) in combination with the data structure of proplets. The computational method of coactivation is *token line intersection*.

As an example, let us assume the A-memory 6.2.3 at a later stage, containing several additional instances of 'bone finding.' They may all be found and coactivated by search patterns which are derived as follows:

6.3.1 Deriving search patterns for a 2nd degree intersection

search patterns		*proplets at now front*
[noun: bone fnc: find prn: K]	⇐	[sur: noun: bone cat: snp sem: indef sg fnc: find ... prn: 97]
[verb: find arg: α bone prn: K]	⇐	[sur: verb: find cat: #n′ #a′ decl sem: pres ind arg: dog bone ... prn: 97]

The patterns (i) contain variables, namely K and α, and (ii) are reduced to the object\predicate relation between find and bone.

As the patterns are moved along their token lines from right to left (CLaTR 4.2.2), they retrieve all find bone pairs of proplets connected by (i) the semantic object\predicate relation and (ii) a shared prn value:

6.3.2 2ND DEGREE INTERSECTION COACTIVATING bone find

member proplets			*now front*	*owners*
sur: **noun: bone** cat: snp sem: indef sg **fnc: find** ... **prn: 23**	sur: noun: bone cat: snp sem: indef sg fnc: eat ... prn: 45	sur: noun: bone cat: snp sem: indef sg fnc: bury ... prn: 66	sur: noun: bone cat: snp sem: indef sg fnc: find ... prn: 97	bone
sur: verb: find cat: #n′ #a′ decl sem: ind past mdd: house ... prn: 11	sur: **verb: find** cat: #n′ #a′ decl sem: ind past **arg: (dog x) bone** ... **prn: 23**	sur: verb: find cat: #n′ #a′ decl sem: ind past arg: chair ... prn: 32	sur: verb: find cat: #n′ #a′ decl sem: ind pres arg: bone ... prn: 97	find

Proplets in the two token lines are subactivated as resonating content if they (i) are in the same bidirectional semantic relation of structure as the two search patterns and (ii) have the same prn value.

Intersections may be of n degrees, $n \geq 2$, depending on the search patterns:

6.3.3 DERIVING SEARCH PATTERNS FOR A 3RD DEGREE INTERSECTION

search patterns		*proplets at now front*
noun: dog fnc: find prn: K	⇐	sur: noun: dog cat: snp sem: def sg fnc: find ... prn: 97
verb: find arg: dog bone prn: K	⇐	sur: verb: find cat: #n-s3′ #a′ decl sem: past ind arg: dog bone ... prn: 97
noun: bone fnc: find prn: K	⇐	sur: noun: bone cat: snp sem: indef sg fnc: find ... prn: 97

The search for resonating contents, partial (6.3.1) or complete (6.3.3) propositions, is a search for counterparts to proplets at the current now front. 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.

6.4 Selective Activation by Autonomous Navigation

Intersections of higher and higher degree result in a narrowing of the resonating contents in the agent's on-board memory. This is suitable for finding fewer and fewer but more and more similar contents resonating with current now front content. The opposite effect is achieved by the third step of the computational pragmatic interpretation in DBS. Called *selective activation*, it is an exploration of wider and wider neighborhoods of a coactivated content.

Selective activation is similar to, but more constrained than, Quillian's (1968) *spreading activation* in that it is restricted to paths along existing semantic relations of structure between pairs of content proplets connected by address. Selective activation has the potential of supplementing the original triggers at the now front with unexpected but relevant information for further reasoning. Like intersection, selective activation is based technically on the database schema of A-memory and the data structure of proplets.

The crucial property for selective activation is the proplet-internal coding of semantic relations by address, making proplets order-free and at the same time connected. Consider the addresses provided by the intersection 6.3.2:

6.4.1 Addresses in connected proplets

proplets	addresses
sur:	
noun: dog	(dog 12)
cat: snp	
sem: indef sg	
verb: find	(find 12)
mdr:	
nc:	
pc:	
prn: 12	
sur:	
verb: find	
cat: #n′ #a′ decl	(find 12)
sem: ind past	
arg: dog bone	(dog 12)(bone 12)
mdr:	
nc: (bury 13)	(bury 13)
pc:	
prn: 12	

For generality, the addresses are shown in extrapropositional format, i.e. with their prn values.

Addresses provide a computational method for establishing semantic relations without any locational requirements such as being in the same formula. For example, the address (dog 12) in 6.4.2 serves as the goal proplet of the intrapropositional think mode traversal operation V↙N, the address (find 12) as the goal proplet of N↗V, the address (bone 12) as the goal proplet of V↘N, and the extrapropositional address (bury 13) as the goal proplet of V→V.

Starting an autonomous navigation from a coactivated proplet in memory is restricted to 'free' semantic relations, i.e. relations which were not involved in establishing the intersection in the first place. As an example consider the following extension of 6.1.2:

6.4.2 Intra- and extrapropositional extension of 6.1.2

sur:	sur:	sur:	sur:	S: yard
noun: dog	adj: **old**	verb: find	noun: bone	T: friday
cat: snp	cat: adnv	cat: #n′ #a′ decl	cat: snp	A: sylvester
sem: def sg	sem: pos	sem: ind past	sem: def sg	R:
fnc: find	mdd: dog	arg: dog bone	fnc: find	3rd:
mdr: **old**	mdr:	mdr:	mdr:	prn: 12
nc:	nc:	nc: **(bury 13)**	nc:	
pc:	pc:	pc:	pc:	
prn: 12	prn: 12	prn: 12	prn: 12	

Assuming the intersection 6.3.3, the above extension has two free semantic relations, namely [mdr: old] in the *dog* proplet and [nc: (bury 13)] in the *find* proplet. While [mdr: old] is intrapropositional, [nc: (bury 13)] is extrapropositional and provides a continuation to a next proposition such as the following:

6.4.3 Extended content: The old dog found a bone. He buried it.

sur:	sur:	sur:	sur:	sur:	sur:	sur:
noun: dog	adj: old	verb: find	noun: bone	noun: (dog 12)	verb: bury	noun: (bone 12)
cat: snp	cat: adnv	cat: #n′ #a′ decl	cat: snp	cat: snp	cat: #n′ #a′ decl	cat: snp
sem: def sg	sem: pos	sem: ind past	sem: def sg	sem: def sg	sem: ind past	sem: def sg
fnc: find	mdd: dog	arg: dog bone	fnc: find	fnc: bury	arg: (d. 12) (b. 12)	fnc: bury
mdr: old	mdr:	mdr:	mdr:	mdr:	mdr:	mdr:
nc:	nc:	nc: (bury 13)	nc:	nc:	nc: (chase 14)	nc:
pc:	pc:	pc:	pc:	pc:	pc:	pc:
prn: 12	prn: 12	prn: 12	prn: 12	prn: 13	prn: 13	prn: 13

Yet another continuation to a third proposition is provided by the [nc: (chase 14)] feature of *bury*. In this way, complex contents in memory consisting of arbitrarily many concatenated propositions may be traversed. This is used not only for providing extended resonating contexts of interpretation, but also for the navigation underlying the narrative speak mode of DBS.

6.5 Literal Pragmatics of Adjusting Perspective

In automatic monitoring, a STAR codes the agent's state at the current moment. However, because a current content of origin recedes inevitably into the past, there are two STARs, the STAR of origin attached to the original 'old' content, called the STAR-0, and the current STAR of the agent's looking back at the old content in memory, called the STAR-1. For example, if the STAR-0 content I see you is retrieved from memory it must be changed to the STAR-1 content I saw you. Also, the values Paris and Thursday of the STAR-0, for example, may change to London and Sunday of the STAR-1, and similarly for the R and the 3rd value.

In communication, the speaker uses the STAR-1 to encode the past STAR-0 content into a language surface. The hearer's interpretation of the speaker's STAR-1 surface necessitates a third STAR, called the STAR-2. For example, the STAR-1 content I saw you must be changed to the STAR-2 content You saw me. The (i) origin, (ii) production, and (iii) interpretation of a content used in language communication are in an obligatory temporal order:

6.5.1 Time line of origin, production, and interpretation

agent A		*agent A*		*agent B*	
STAR−0	*conversion−1*	STAR−1	*conversion−2*	STAR−2	
·····	·····	·····	·····	·····	► *time line*
origin		production		interpretation	
I see you.		I saw you.		You saw me.	

The STAR-0 values Paris and Friday, for example, may change into the STAR-1 values London and Sunday, which in turn may change to New york and Wednesday of the STAR-2, and similarly for the R and the 3rd value.

To illustrate the changes from a STAR-0 content of origin to a STAR-1 content of production to a STAR-2 content of interpretation in more detail, let us spell out the above examples as sets of proplets concatenated by address:

6.5.2 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[7]
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:
mdr:	mdr:	mdr:	prn: 12
nc:	nc:	nc:	
pc:	pc:	pc:	
prn: 12	prn: 12	prn: 12	

By definition, a STAR-0 content is (i) without language (quasi language-independent) and (ii) 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 at least six STAR-1 variants which differ semantically, but are pragmatically equivalent to the STAR-0 content 6.5.2:

6.5.3 Pragmatically equivalent STAR-1 variants

STAR-1 A: Sylvester remembers the content 6.5.2 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 E and F, past STAR-0 values which were overwritten by the current ones are preserved by writing them into the content.[8]

The hearer's STAR-2 counterparts to the STAR-1 variants are as follows:[9]

6.5.4 Pragmatically equivalent STAR-2 variants

STAR-2 A: Does not exist for nonlanguage content.
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 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 pronominal indexicals. This is because the interpretation of pronominals in the speaker's conversion-1 is the same as the STAR-0 content, but inverted in the hearer's conversion-2.

[7] Using weekdays as T values is crude as compared to nano-seconds, but sufficient for the following examples.
[8] For formal details see CC, Sect. 7.1.
[9] For formal details see CC, Sect. 8.1.

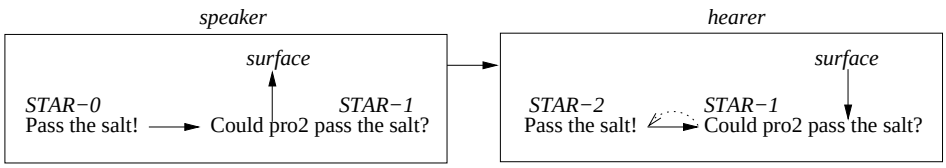
The pragmatic equivalence of the semantically different contents 6.5.2, 6.5.3, and 6.5.4 is based on the choice between coding certain values either in the content (as concepts or names) or as STAR values. Furthermore, there is a choice between leaving the S and T values of the STAR-0 implicit, as in variants A-D in 6.5.3 and B-D in 6.5.4, or writing them explicitly into the content, as in the variants E and F. In contents which are semantically generic, such as Water is H_2O, STAR values make no difference pragmatically.

6.6 Nonliteral Pragmatics of Syntactic Mood Adaption

The examples analyzed in the previous section are from literal pragmatics insofar as the semantic differences between STAR-0, STAR-1, and STAR-2 contents are pragmatically *compensated*. In nonliteral pragmatics, in contrast, nonliteral STAR-1 contents in the speak mode must be *reverted* in the hear mode back to the literal STAR-0 contents[10] in order for communication to succeed.

A classic example (Austin [1955]1962) of nonliteral pragmatics is syntactic mood adaptation:

6.6.1 Syntactic mood adaptation IMP-INT



The speaker's communicative purpose of the STAR-1 STAR-2 conversion is softening a command (imperative[11]) into a polite request (yes-no interrogative). If the hearer were to take the speaker's STAR-1 content literally by answering yes or no, communication would fail. Syntactic mood adaptations are based on inferences which are used deductively in the speak mode and abductively, i.e. in the opposite direction (CC Sects. 7.3, 8.3), in the hear mode.

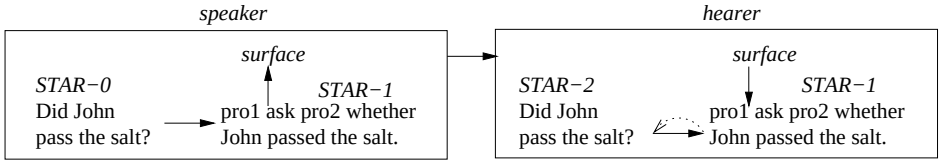
Syntactic mood adaptions are common in the European languages, but not universal.[12] The following two examples are based on the inferences INT-DECL (CC 7.4.2) and IMP-DECL (CC 7.5.2), respectively.

[10] Modulo the obligatory hear mode adjustments of a STAR-2 content.

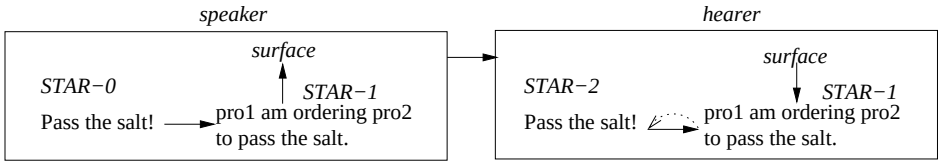
[11] The grammatical subject of imperatives in English and similar languages is implicit: it is automatically assumed to be pro2 in the speak mode and pro1 in the hear mode, without any surface manifestation. Consequently, the hearer's standard STAR-2 reversal from pro2 to pro1 is implicit as well.

[12] For example, in Korean the corresponding rhetorical purpose is alternatively coded in the morphology of honorifics. Thanks to Prof. Kiyong Lee.

6.6.2 SYNTACTIC MOOD ADAPTATION INT-DECL



6.6.3 SYNTACTIC MOOD ADAPTATION IMP-DECL



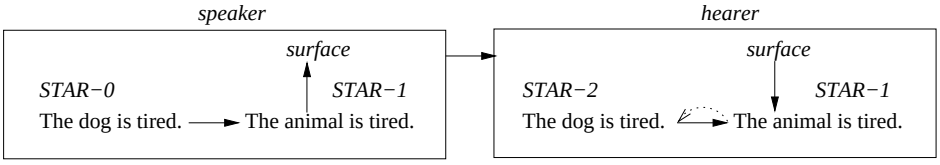
The extensions provided by the deductive use intensify the speaker's content and are known as 'explicit performatives' in Speech Act Theory. There are more syntactic-mood adaptations, such as the interrogative-imperative adaptation from Did you pass the salt? to Tell me if you passed the salt![13]

6.7 Nonliteral Pragmatics of Figurative Use

Syntactic mood adaptation and figurative use are alike in that (i) the same inference is employed deductively in the speak mode and abductively in the hear mode and (ii) the hearer is required to reconstruct the speaker's STAR-0 content. They differ in that the inferences of syntactic mood adaptation take complete propositions as input and output while the inferences of figurative use take only an elementary or phrasal part.

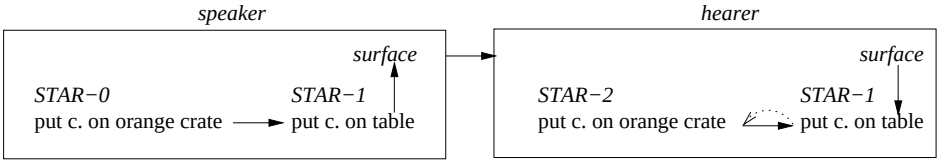
Figurative use is subject to the invariance constraint (CC 6.4.1), according to which the figurative replacement must be of the same syntactic category and the same semantic field as the literal original. The basis of inferencing are lexical relations such as hyponymy, metonymy, property sharing (Glucksberg 2001 p. 58), and abbreviation. Consider the following examples (see CC Chap. 9 for details):

6.7.1 FIGURATIVE USE BASED ON HYPONYMY RELATION

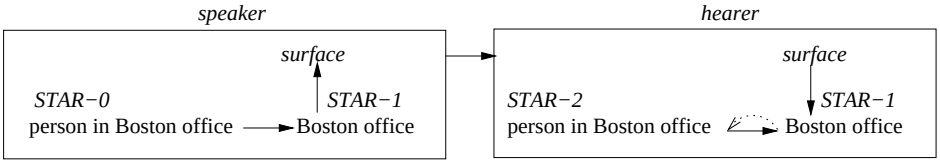


[13] The dependence of nonliteral use pragmatics on specific pronouns, concepts, tense, mood, etc. is reminiscent of construction grammar (Fillmore 1988).

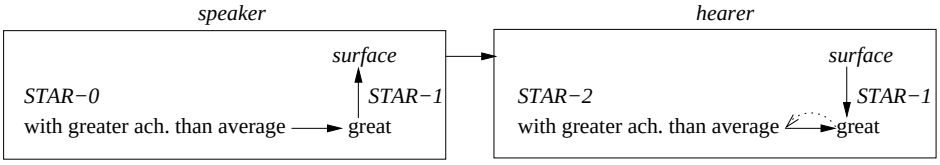
6.7.2 Figurative use based on shared property inference



6.7.3 Figurative use based on metonymy relation[14]



6.7.4 Speak and hear mode of an abbreviating adnominal use



As in all abductive use, there is no certainty regarding the output of the inference (i.e. of the antecedent). For example, if the orange crate in 6.7.1 were accompanied by a footstool, a sideboard, 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.

Conclusion

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 on-board orientation system, i.e. the STAR, are tokens and belong to the pragmatics, while contents unconnected to a STAR are types and belong to the semantics.

The relevant parameter values of the agent's on-board orientation system are continuously recorded in the form of STAR proplets which specify the

[14] Hobbs et al. 1993.

agent (pro1) and monitor the agent's moment by moment location in time and space as well as 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.

Bibliography

Austin, J.L. ([1955]1962) *How to Do Things with Words*, Oxford: Clarendon Press

CC = Hausser, R. (2019) *Computational Cognition; Integrated DBS Software Design for Data-Driven Cognitive Processing*, `lagrammar.net`

CLaTR = Hausser R. (2011) *Computational Linguistics and Talking Robots; Processing Content in DBS*, pp. 286. Springer (preprint 2nd ed. at `lagrammar.net`)

Cresswell, M. (1972) "The world is everything that is the case," *Australasian Journal of Philosophy*, Vol. 50:1–13.

Elmasri, R., and S.B. Navathe (2017) *Fundamentals of Database Systems, 7th ed.* (1st ed. 1989), Redwood City, CA: Benjamin-Cummings

Fillmore, C.J. (1988) "The mechanisms of Construction Grammar," *Proceedings of the Fourteenth Annual Meeting of the Berkeley Linguistics Society (1988)*, pp. 35–55

Hobbs, J.R., M.E. Stickel, D. Appelt, and P. Martin (1993) "Interpretation as abduction," *Artif. Intell.* Vol. 63:69–142

Horn, L., and G. Ward (eds.) (2004) *The Handbook of Pragmatics*, Oxford: Blackwell Publishers

Kempson, R. (2001) "Pragmatics: language and communication," Chapter 16 in *The Handbook of Linguistics*, Aronoff, M. and J. Rees-Miller (eds.), Oxford: Blackwell Publishers, pp. 394–427

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, pp. 221–242

Peirce, C.S. (1931–1935) *Collected Papers*. C. Hartshorne and P. Weiss (eds.), Cambridge, MA: Harvard Univ. Press

Quillian, M. (1968) "Semantic memory," in M. Minsky (ed.), *Semantic Information Processing*, Cambridge, MA: MIT Press, pp. 227–270

7. Discontinuous Structures in DBS and PSG

April 24, 2020

Abstract

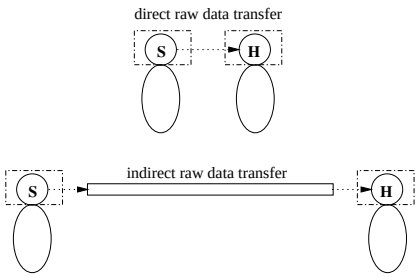
In linguistics, grammatical constructions with semantically related word forms are called *discontinuous* if the word forms are not adjacent. For example, in (i) Peter looked the number up., the semantically related looked ... up are separated by the_number and in (ii) Yesterday Mary danced. the semantically related Yesterday ... danced are separated by Mary.

This paper explores why example (i) poses a descriptive problem for PSG (Phrase Structure Grammar), but not for DBS (Database Semantics), and why example (ii) poses a descriptive problem for DBS, but not for PSG. Then the PSG and DBS proposals for solving their respective problem are compared.

7.1 The Time-Linear Structure of Natural Language

Many syntactic-semantic content structures in human cognition are hierarchical, but the transfer of content in language communication is strictly linear. In the medium of speech, this holds for direct language communication, such as talking face to face, and indirect language communication, such as talking on the phone:

7.1.1 Comparing face-to-face with on the phone talking



The cognitive aspects are located inside the dotted boxes around the agents' heads. They operate the same regardless of whether communication is face to face (direct raw data transfer) or talking on the phone (indirect raw data transfer). It is therefore not appropriate to build linguistic semiotics on information theory, as has been proposed by Eco[1] (1975).

All that is required of an artificial or natural transfer channel is the transmission of data without distortion (Shannon and Weaver 1948). However, though the transfer channel is not the place for reconstructing cognition, it poses a

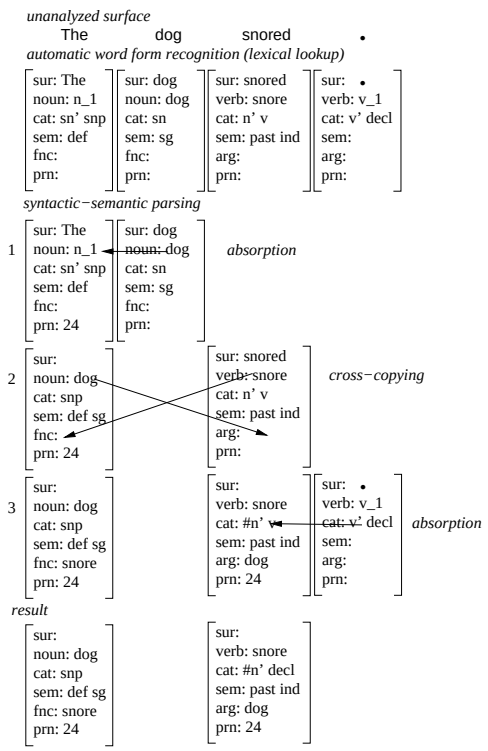
[1] Eco's most basic prototype of communication is a buoy "telling" an engineer the water level of a lake; the engineer's partner of discourse is mother nature. Grice's (1957) "bus bell model" and Dretsky's (1981) "doorbell model" have another human ringing the bell (CLaTR 2.2.4).

crucial structural requirement for language communication: the signs must be in a strictly linear order (canonized by de Saussure [1916](1972) as his *second principe*). This is because humans can neither produce nor interpret two or more words, phrases, or sentences simultaneously. Thus the structures processed by cognition may be as hierarchical as necessary as long as language contents are coded in time-linear order fit for the transfer channel.

This requires a suitable data structure in the sense of computer science. In DBS, it is non-recursive feature structures with ordered attributes, called proplets (as the smallest units of propositions). A content is defined as a set (order-free) of proplets connected by the classical semantic relations of structure coded by address. The speak mode takes a content as input and produces a surface sequence as output by traversing the input content along the address-coded semantic relations between proplets. The hear mode takes a surface sequence as input and produces an output content by concatenating proplets by address into a set. Because of this difference in the respective inputs and outputs, the software for the agent's speak and hear modes can not be the same.

As an example of the hear mode, consider the surface The dog snored. taken as input and the content produced by the derivation as output:

7.1.2 Deriving content from surface in Hear mode



The input is unconnected lexical proplets (nonempty sur values). The output content is connected nonlexical proplets (empty sur values). The derivation order is (i) *time-linear* (left-associative), shown by the stair-like addition of next word proplets, the analysis is (ii) *surface compositional* because each lexical item has a concrete sur value and there are no surfaces without a proplet analysis, and the activation and application of operations is (iii) *data-driven*.

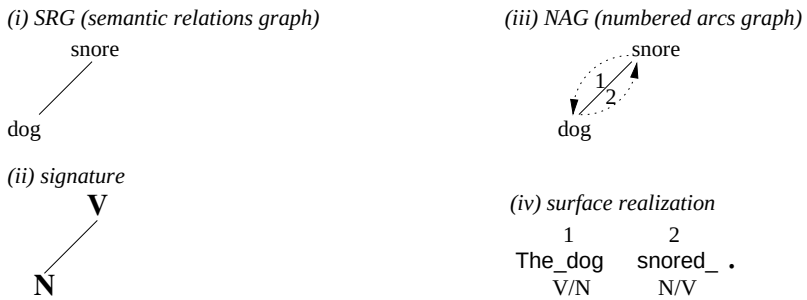
As sets of proplets connected by address, contents are suitable for storage in and retrieval from the agent's onboard memory (content-addressable database):

7.1.3 Content of The dog snored.

sur:	sur:
noun: **dog**	verb: **snore**
cat: def sg	cat: #n′ decl
sem:	sem: past ind
fnc: **snore**	arg: **dog**
mdr:	mdr:
nc:	nc:
pc:	pc:
prn: 24	prn:24

Semantically, the order-free proplets implement the subject/predicate relation. The semantic relation of structure may be shown graphically as follows:

7.1.4 Semantic relations underlying speak mode



The *(i) SRG* and the *(ii) signature* on the left show the static aspect of the hierarchical content structure. The *(iii) NAG (numbered arcs graph)* and *(iv) surface realization* on the right show its dynamic traversal.

The arc numbers of the *NAG* are used for specifying (1) a think mode navigation and (2) a think-speak mode surface production as shown by the *surface realization*. The main intrapropositional traversal operations in the think mode of DBS are (1) predicate↙subject, (2) subject↗predicate, (3) predicate↘object, (4) object↖predicate, (5) noun↓adnominal, (6) adnominal↑noun, (7) verb↓adverbial,

(8) adverbial↑verb, (9) noun→noun, (10) noun←noun, (11) verb→verb, (12) verb←verb, (13) adnominal→adnominal, and (14) adnominal←adnominal.

The *(iv) surface realization* shows the language-dependent production. It is implemented as the speak mode riding piggy-back on the think mode navigation. The concepts of dog and snore are provided as types by the agent's memory as a *declarative specification* and as tokens by the *operational implementation* of the agent's interface component.

7.2 Constituent Structure Paradox of PSG

DBS and PSG differ ontologically in that DBS is data-driven agent-based, while PSG is substitution-driven sign-based. The derivations of PSG generate different language expressions from the same node, usually called S for sentence or start. The number of possible tree structures for a given surface grows exponentially with the length of the surface.

Consider two different PSG trees for a given three-word sentence:

7.2.1 Two different PSG trees for same unambiguous surface

From a formal point of view, both trees are equally well-formed.

However, because more than one tree for an unambiguous surface does not make sense linguistically, there must be an intuitive principle for choosing the "good" one. In chomskyan linguistics, this is the principle of Constituent Structure:[2]

7.2.2 Definition of Constituent Structure

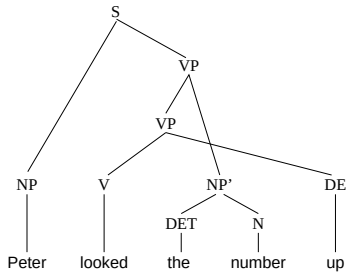
1. Words or constituents which belong together semantically must be dominated directly and exhaustively by a node.
2. The lines of a Constituent Structure may not cross (*nontangling condition*).

[2] The notion of Constituent Structure is quite recent. It evolved from the *immediate constituent analysis* of L. Bloomfield (1887–1949). Bloomfields' student Z. Harris (1909–1992) turned the constituent analyses into substitution and movement tests. Harris's student N. Chomsky turned the methodologically motivated tests into generative rules, called transformations and offered as innate.

Assuming that knows and John (predicate-object) belong closer together semantically than Julia and knows (subject-predicate), the tree on the left satisfies 7.2.2 and is therefore considered linguistically correct, while the tree on the right is not. Yet it has been known at least since 1953 (Bar-Hillel 1964, p. 102) that there are certain natural language constructions with "discontinuous elements" which violate the definition of Constituent Structure.

Known as the Constituent Structure Paradox (FoCL Sect. 8.5), the problem may be illustrated with discontinuous look__up in the following attempts to analyze Peter looked the number up as a Constituent Structure:

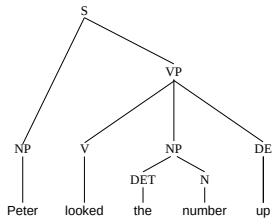
7.2.3 Violating the second condition of 7.2.1



Here the semantically related expressions looked and up are dominated directly and exhaustively by a node, satisfying the first condition of 7.2.1. The analysis violates the second condition, however, because the lines cross.[3]

The alternative attempt satisfies the second condition, but violates the first:

7.2.4 Violating the first condition of 7.2.1



Here the lines of the tree do not cross, satisfying the second condition, but the semantically related expressions looked – up, or rather the nodes V and DE dominating them, are not *exhaustively* dominated by a node. Instead, the node directly dominating V and DE also dominates the NP the_ number.

Despite this empirical problem with definition 7.2.2 Chomsky continued to use Constituent Structure for linguistically motivating the choice between the

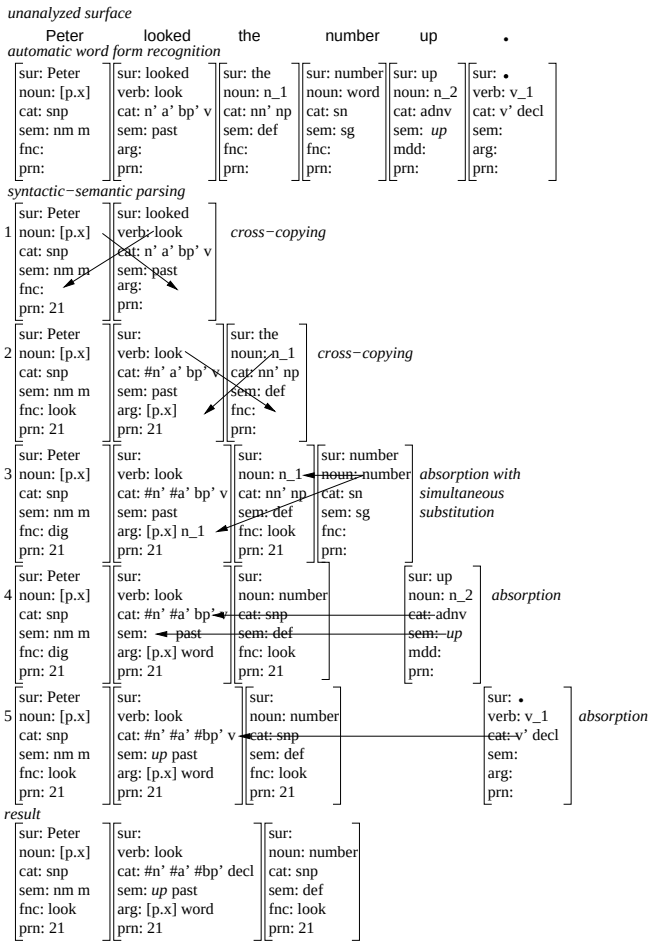
[3] DPSG (Discontinuous Phrase Structure Grammar, Bunt et al. 1987) argues for accepting crossing lines in Phrase Structure Trees. This was preceded by pleas for using only context-free phrase structure by Harman 1963, McCawley 1982, Gazdar et al. (1985), and others.

multiplicity of possible phrase structure trees for unambiguous English surfaces (7.2.1). To accommodate the natural surfaces, he added a transformation (movement) component on top of the context-free PSG base.

The output of the base satisfies Constituent Structure but disregards some natural surfaces. The output of the transformations satisfies the natural surfaces, but sometimes violates Constituent Structure. The introduction of transformations raised the computational complexity degree n^3 (polynomial) of context-free PSG to n^∞ (undecidable) of Transformational Grammar (Peters&Ritchie 1973).

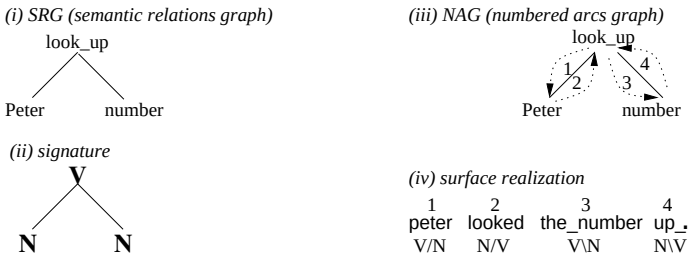
In DBS, the example shown in 7.2.3 and 7.2.4 for PSG derives without any problem in the strictly time-linear, surface-compositional, data-driven manner of the hear mode:

7.2.5 Graphical hear mode derivation in DBS



The semantic relations in the resulting content are shown by the following graph analysis. The surface realization of the speak mode is shown for English:

7.2.6 Graph analysis underlying production in DBS



In the content derived in 7.2.5, the discontinuous element up is coded as the initial sem value of the *look* proplet in line 4. In the *(iv) surface realization*, up_• is realized from the finite verb (goal proplet of arc 4 in the *(iii) NAG*).

7.3 Suspension in DBS

The reason why Peter looked the number up violates Constituent Structure, but Yesterday Mary danced does not may be shown by comparing their respective context-free PSG derivations, based on possible substitutions:

7.3.1 Context-free PSGs for two discontinuous structures

Peter looked the word up

S → NP VP
VP → VP NP
VP → V NP DE
NP → N
NP → DET N
N → Peter
V → looked
DET → the
N → number
DE → up

Yesterday Mary danced

S → ADV VP
VP → NP V
NP → N
ADV → yesterday
N → Mary
V → danced

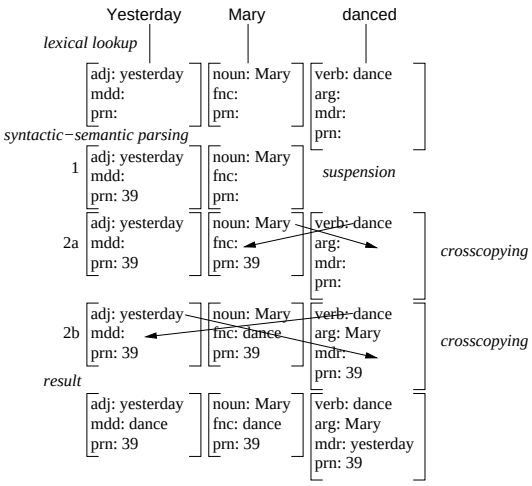
The PSG for the variant 7.2.4 on the left violates the requirement of *exhaustive* dominance with the rule VP→V NP DE.[4] The PSG on the right, in contrast, generates Yesterday Mary danced without any problem: first the ADV is

[4] The variant 7.2.3 requires a movement transformation in order to satisfy condition 1 of the Constituent Structure definition 7.2.2. There is, however, the context-free PSG analysis (7.5.1 on the left) which satisfies 7.2.2, but may still be rejected on grounds of linguistic intuitions in PSG.

placed in initial position by S→ADV VP; then VP→NP V places the subject noun after the ADV and before the V (7.5.1 on the right).

The apparent problem of DBS with Yesterday Mary danced results from the time-linear derivation order. It creates a temporary situation in which the elementary adverb yesterday can not be connected because the semantically related verb has not yet arrived. The solution is a *suspension* until danced becomes available (derivations 7.3.2–7.3.5 in 'convenient notation').

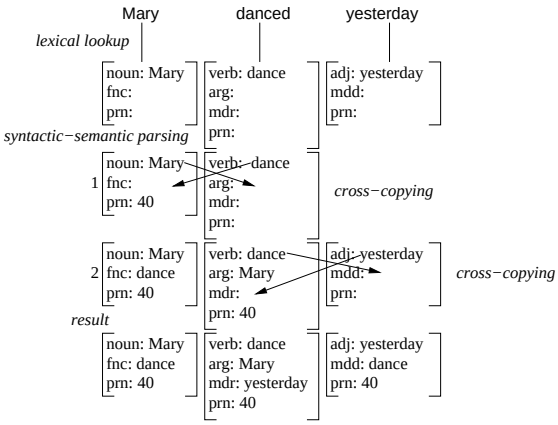
7.3.2 HEAR MODE DERIVATION OF Yesterday Mary danced.



Because operations are data-driven by the input, instances of suspension are handled automatically, without any need for extra software.

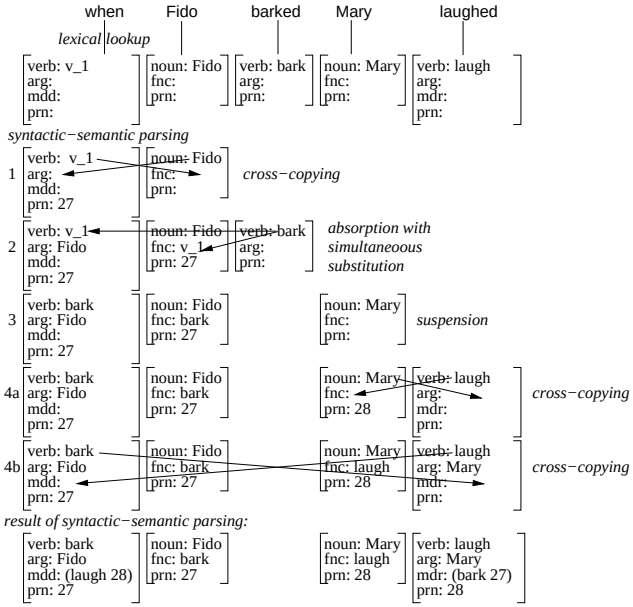
The phenomenon is asymmetric because the semantically equivalent word order Mary danced+yesterday requires neither suspension nor absorption.

7.3.3 HEAR MODE DERIVATION OF Mary danced yesterday.



Suspension at the elementary level scales up directly to the phrasal and clausal levels. At the clausal level, the relation is extrapropositional:

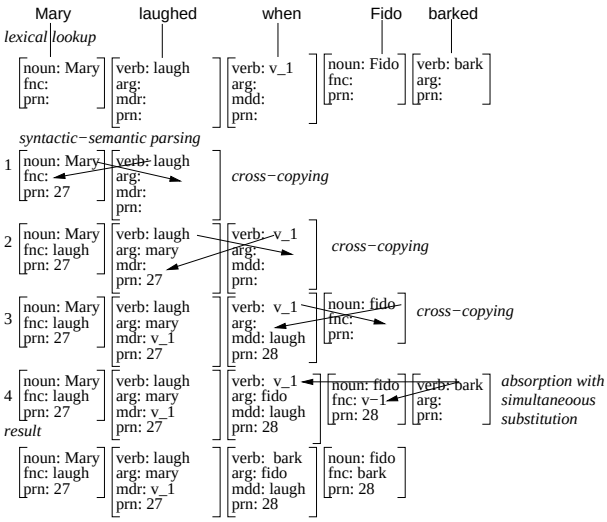
7.3.4 INTERPRETATION OF When Fido barked Mary laughed



The suspension occurs in line 3 and is compensated in lines 4a and 4b.

If the optional modifier clause follows the main clause, there is no suspension, just as in 7.3.3:

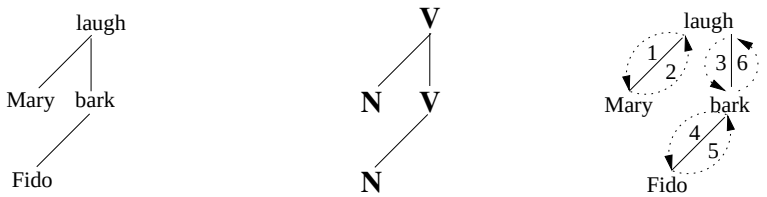
7.3.5 INTERPRETATION OF Mary laughed when Fido barked



Consider the semantic relations graph which underlies both surfaces:

7.3.6 Semantic relations underlying speak mode production

(i) SRG (semantic relations graph) *(ii) signature* *(iii) NAG (numbered arcs graph)*



(iv) surface realization

	1	2	3	4	5	6
(a)	Mary	laughed	when	Fido	barked	•
	V/N	N/V	V|V	V/N	N/V	V|V

	3	4	5	6	1	2	
(b)	When	Fido	barked		Mary	laughed	•
	V|V	V/N	N/V	V|V	V/N	N/V	

The suspension at the elementary level 7.3.2 and its variant 7.3.4 at the clausal level of grammatical complexity show that there are natural surface orders in which suspension cannot be avoided. This is benign insofar as it does not go above the linear complexity of DBS (TCS).

7.4 Discontinuity with and without Suspension in DBS

The domain-range structure of the semantic relations of structure in natural language may be viewed as a constellation of filler and slot, with the domain providing the filler (also called the argument, actant, or complement) and the range (also called functor, codomain, or valency[5] carrier) providing the slot. In (i) subject/predicate, the subject is the filler and the predicate provides the slot, in (ii) object\predicate the predicate provides the slot and the object is the filler, and in (iii) modifier|modified the modifier is the filler and the modified provides the slot. In (iv) coordination, conjunct n provides the slot and conjunct n+1 is the filler.[6]

If filler and slot are adjacent in the hear mode, i.e. if there are no intervening items, order does not make a difference. However, if there are intervening items between filler and slot, and the slot precedes, then (a) the slot defines the kind of compatible filler, (b) the relation is initiated, and (c) the hearer can

[5] L.Tesnière (1959).

[6] This explicit specification of the semantic relations of structure is in stark contrast to the rather vague notion of "belonging semantically together" of the Constituent Structure definition 7.2.2. Also, the filler-slot distinction is missing in PSG.

simply wait until the filler arrives and plops into the slot. If the filler precedes, in contrast, there may be many kinds of slot which is why no relation can be initiated; instead the hearer must wait until automatic word form recognition provides the slot, finally enabling an on the spot filler-slot combination.

In English filler-slot constellations, the intervening items must often[7] be bridged by a suspension. Consider the following examples:

7.4.1 Suspension in filler-slot discontinuities

1. Clausal subject precedes main clause.
   That Fido found a bone surprised Mary.
2. Adverbial precedes predicate (7.3.2)
   Perhaps Fido ... is sleeping...
3. Clausal modifier precedes main clause (7.3.4)
   When Fido barked Mary laughed.
4. Long distance dependency (Sect. 5.7.1)
   Who(m) did John say that Bill believes that Mary loves?

In many slot-filler constructions, the intervening word forms may be bridged by *absorption*:

7.4.2 Absorption in slot-filler discontinuities

1. Period precedes subject in extrapropositional coordination (TExer3 Sect. 2.1):
   Mary sleeps. Fido snores.
2. Verb precedes bare preposition (TExer3, Sect. 2.6)
   Peter looked the number up.
3. Clausal object (TExer3, Sect. 2.6)
   Mary heard that Fido barked.
4. Main clause precedes clausal modifier (7.3.5)
   Mary laughed when Fido barked.
5. Repeating object clauses (TExer3, Sect. 5.6)
   Mary saw the man who loves the woman who fed Fido.

In gapping constructions, however, the relation between the order of filler and slot, and the use of suspension vs. absorption seems to be in reverse as compared to 7.4.1 and 7.4.2:

7.4.3 Absorption vs. suspension in gapping

1. Subject gapping (TExer3, Sect. 5.2)
   Bob bought an apple, peeled a pear, and ate a peach.
   Exception: filler-slot without suspension
2. Predicate gapping (TExer3, Sect. 5.3)
   Bob bought an apple, Jim a pair, and Bill a peach.
   Exception: filler-slot without suspension

[7] In That Fido barked amused Mary (TExer3, Sect. 2.5) the V/V relation between *bark* and *amuse* is not discontinuous and no suspension is needed. It is similar in the V|A relation of Mary danced yesterday (7.3.3).

3. Object gapping (TExer3, Sect. 5.4)
   Bill bought, Jim peeled, and Bill ate a peach.
   Exception: slot-filler with suspension

Object gapping is special in that it requires not only suspension, but also a derivation-external cache until the filler (as the standard location of the gap list) arrives (TExer3 Sect. 5.4).

7.5 Conclusion

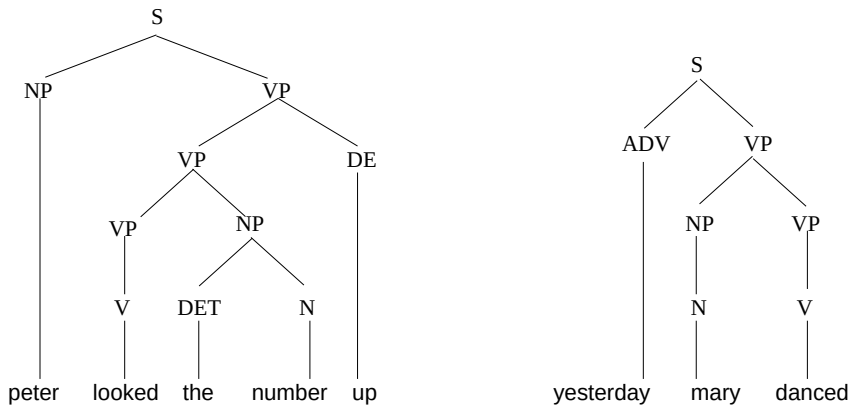
This paper started with an enigma. The examples

(i) Peter *looked* the number *up*. and

(ii) *Yesterday* Maria *danced.*

are both discontinuous. For context-free PSG, (i) poses a problem, solved by adding a movement component, while (ii) runs straight through. For DBS, (i) runs straight through, while (ii) poses a problem, solved by suspension (7.3.2). Let us conclude by showing what applying the PSG solution for example (ii) (7.3.1) to example (i) would look like:

7.5.1 Phrase structure of example (i) in concord with 7.2.2



If the phrase structure on the right satisfies the Constituent Structure definition 7.2.2 then the one on the left does as well. So what is so wrong with the left phrase structure that adding a movement component was preferred even at the price of making the grammar undecidable?[8]

[8] Actually, discontinuities were not the main motivation for introducing a movement component. Instead it was the assumption of a universal, innate context-free base in combination with a transformation component (nativism). Thus a discontinuity analysis exceeding the generative power of context-free PSGs came in handy as "empirical" support for the Standard Theory (Chomsky 1965). The undecidability of Transformational Grammar was proven in active consultation with Chomsky.

The lines do not cross in either tree. Also the rules (a) VP → VP DE used on the left and (b) S → ADV VP used on the right are alike in that two nodes which "belong together semantically" are dominated directly and exhaustively by a single node. The rules differ, however, in what they relate. Even though the Discontinuous Element DE is a function word, i.e. up, and the VP is a content word, i.e. looked, they are treated by rule (a) as if they were separate, equal entities. The nodes connected by rule (b), in contrast, are the content words yesterday and danced.

Treating a function word such as the bare preposition up as an equal partner to a content word may be rejected on grounds of linguistic intuition. It is of a similar nature as the intuition that predicate and grammatical object belong closer together semantically than subject and predicate[9](7.2.1).

Acknowledgement: Thanks to Professor Kiyong Lee for helpful comments.

Bibliography

Bar-Hillel, Y. (1964) *Language and Information. Selected Essays on Their Theory and Application*, Reading, Mass.: Addison-Wesley

Bloomfield, L. (1933) *Language*, New York: Holt, Rinehart, and Winston

Bunt, H., J. Thesingh, and K. van der Sloot (1987) "Discontinous Constituents in Trees, Rules, and Parsing," in *Proc. 3rd Conference of the European chapter of the ACL,* pp. 203-210. Copenhagen, DK, Univ. of Copenhagen

CC = Hausser, R. (2019) *Computational Cognition; Integrated DBS Software Design for Data-Driven Cognitive Processing*, DOI: 10.13140/RG.2.2.33658.-85440, `lagrammar.net`

Chomsky, N. (1965) *Aspects of the Theory of Syntax*, Cambridge, Mass.: MIT Press

CLaTR = Hausser, R. (2011) *Computational Linguistics and Talking Robots; Processing Content in DBS*, pp. 286. Springer (preprint 2nd ed. `lagrammar.net`)

[9] The DBS counterpart to this intuition in English is the asymmetry between the subject/predicate and the object\predicate relation in that the filler precedes the slot in one and follows in the other.

Dretske, F. (1981) *Knowledge and the Flow of Information*, Cambridge, Mass.: Bradford Books/MIT Press

Eco, U. (1975) *A Theory of Semiotics*, Bloomington, Indiana: Indiana Univ. Press

FoCL = Hausser, R. (1999) *Foundations of Computational Linguistics*, 3rd ed. 2014, Springer

Gazdar, G., E. Klein, G. Pullum, and I. Sag (1985) *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard Univ. Press

Grice, P. (1957) "Meaning," *Philosophical Review*, Vol. 66:377–388

Harman, G. (1963) "Generative Grammar without Transformational Rules: a Defense of Phrase Structure," *Language*, Vol. 39:597–616

Harris, Z. [1941](1951) *Methods in Structural Linguistics*, Chicago: Univ. of Chicago

McCawley, J.D. (1982) "Parentheticals and Discontinuous Constituent Structure," *Linguistic Inquiry*, Vol. 13.1:91-106

NLC = Hausser, R. (2006) *A Computational Model of Natural Language Communication: Interpretation, Inference, and Production in DBS*, pp. 360. Springer (preprint 2nd ed. `lagrammar.net`)

Peters, S., and R. Ritchie (1973) "On the Generative Power of Transformational Grammar," *Information and Control*, Vol. 18:483–501

Saussure, F. de [1916](1972) *Cours de linguistique générale*, Édition critique préparée par Tullio de Mauro, Paris: Éditions Payot

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

Tesnière, L. (1959) *Éléments de syntaxe structurale*, Paris: Editions Klincksieck

TExer3 = Hausser, R. (2020) *Twentyfour Exercises in Linguistic Analysis, DBS software design for the Hear and the Speak mode of a Talking Robot* (3rd ed. `lagrammar.net`)

8. Syllogisms as Computational Inferences

Sept. 30, 2019

Abstract

The classical categorical syllogisms originated and evolved in times without computers, sensors for vision and audition, and actuators for manipulation and vocalization. Therefore, a substitution-driven sign-based ontology was the only practical option.

That it is nevertheless possible to translate categorical syllogisms into the data-driven agent-based inferencing of DBS is because they use the same set-theoretic structures. The following reconstruction proceeds from the diagrams by Swiss mathematician Leonard Euler, used in the year 1761 in a famous "letter to a princess."

keywords: sign-based vs. agent-based; logical vs. common sense reasoning; categorical syllogisms; categorical judgements; Euler diagrams

8.1 Logical vs. Common Sense Reasoning

A basic distinction in analytic philosophy is between logical reasoning and common sense reasoning. Logical reasoning is based on set theory, which is why the associated inferences in DBS are called S-inferences. Common sense reasoning, in contrast, is without a set-theoretic aspect and the associated inferences are called C-inferences in DBS.

In the human prototype, S-inferences and C-inferences are not separated, but work smoothly together. Therefore, the computational model of reasoning in DBS uses the same general inference schema and the same data structure for S- and C-inferences. Consider the following comparison of the two kinds of DBS inferences as schematic examples:

8.1.1 Example of an S-inference (FERIO)

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

8.1.2 Example of a C-inference (CAUSE_and_EFFECT)

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

Each inference binds the subject term of the input to the variable α in the antecedent of the pattern, which enables the consequent to derive the output.

The implementation of the syllogism FERIO as an S-inference is illustrated in 8.6.10, while the corresponding implementation of the C-inference 8.1.2 is shown as the following software operation:

8.1.3 Applying the C-inference 8.1.2 in DBS format

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

The content and the pattern level consist of nonrecursive feature structures with ordered attributes. Called proplets[1], they serve as the computational data structure of DBS. The proplets of a content are order-free but connected by the classical semantic relations of structure, i.e. functor-argument and coordination, coded by address.

The proplets at the pattern level use variables as values for the ‘core’ and the ‘continuation’ attributes, those at the content level have corresponding constants. The content proplets of the antecedent serve as input to the inference by matching and binding their constants to the corresponding variables of the pattern proplets, which enables the consequent to derive the output.

For computational pattern matching to be successful (i) the attributes of the pattern proplet must be a *sublist*, (ii) the variables of the pattern proplet must be *compatible*, and (iii) the constants of the pattern proplet must be *identical* with those of the corresponding content proplet directly underneath. By binding the variables of the antecedent to the constants of the input, the consequent derives the output.

[1] Because they are the elementary items of propositions.

S-inferences and C-inferences differ in the source of their reasoning. For example, in the S-inference 8.1.2, the source is the disjunction between the concepts homework and be_fun and the intersection between reading and homework (8.6.8), which are assumed to be generally accepted. In the C-inference 8.1.3, in contrast, the source is something observed by the agent(s).[2]

8.2 Categorical Syllogisms

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.[3] In the modern era, the syllogisms have been based on the intuitions of set theory.

A categorical[4] syllogism consists of three parts, called premise 1, premise 2, and the conclusion. This may be shown schematically as follows:

8.2.1 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:

8.2.2 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

[2] The resulting set-theoretic relation between being cranky and being hungry in 8.1.2, i.e. intersection, is merely a consequence of the reasoning, and not the source.

[3] 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 Yinsheng and Qiao Xiaodong (2009).

[4] 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.

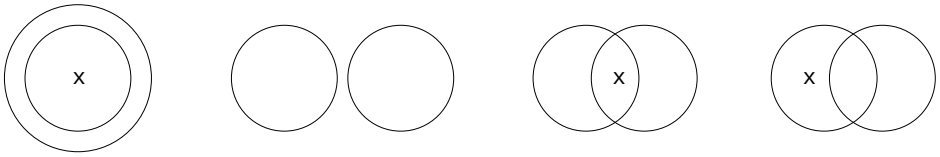
The first-order Predicate Calculus representation in the third column is in a linear notation called *prenex normal form*, which superseded Frege's (1879) graphical format.

The four categorical judgements combine into 256 (2^8) possible syllogism, of which 24 have been found valid. The syllogisms reconstructed in this paper as DBS inferences are BARBARA,[5] CELARENT, DARII, FERIO, BAROCO, and BOCARDO, plus the modi ponendo ponens and tollendo tollens as special[6] cases.

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

8.2.3 Set-theoretic counterparts of categorial judgements

A	E[7]	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[8] (Euler 1761), the set-theoretic constellations are used in DBS to reconstruct the valid syllogisms as data-driven, agent-based S-inferences. As an example, consider the schematic application of modus Barbara in DBS:

8.2.4 Modus Barbara as a DBS inference

inference: α be_human $\Rightarrow$ α be_mortal

$\Uparrow$ $\Downarrow$

input: all Greeks be_human output: all Greeks be_mortal

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 common

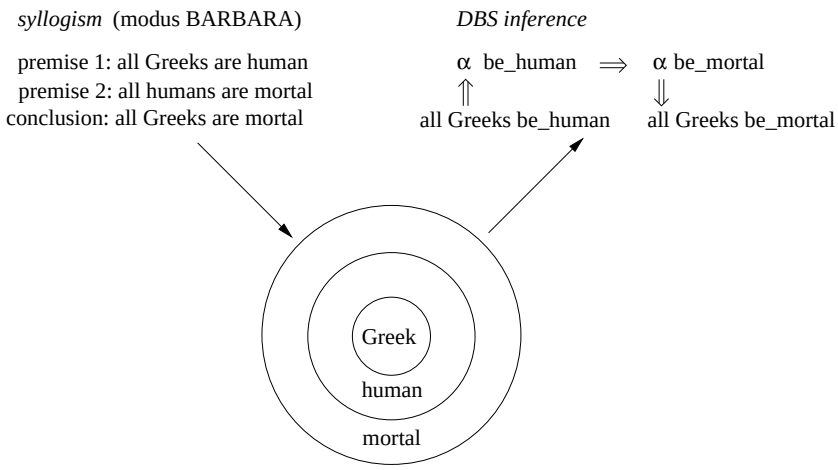
[5] The scholastics used the vowels of the categorical judgements in the names of the associated syllogisms as mnemonic support. For example, the three vowels in the name of modus BARBARA indicate that the categorical judgements of the propositions serving as the two premises and the consequent are all of the kind **A**, i.e. universal affirmative (8.2.2).

[6] The modi ponendo ponens (8.3.1) and tollendo tollens (8.4.1) 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 proper, for example BARBARA or FERIO.

foundation of both the sign-based classical syllogisms and their agent-based counterparts in DBS.

The transition from a categorial syllogism to a DBS inference is shown by the following example using modus BARBARA:

8.2.5 From syllogism to DBS inference



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, 8.4.4), (2) a subset, e.g. some pets (particular, 8.6.4), or (3) an element, e.g. Socrates (individual, 8.6.3). In the consequent, the input-binding of α derives the output.

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:

8.2.6 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.

---

[8] 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.

[8] 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 (8.6.8).

The universal version is modeled after modus BARBARA (8.5.1), the particular version after modus DARII (8.6.1), and the individual version after modus ponendo ponens (8.3.1).

The second triple negates the consequent:

8.2.7 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 (8.5.6), the particular version after modus FERIO (8.6.6), and the individual version after modus tollendo tollens (8.4.1).

The third triple negates the antecedent:

8.2.8 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:

8.2.9 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.

8.3 Modus Ponendo Ponens

Modus ponendo[9] ponens serves as the individual version of 8.2.6. The standard representation in Predicate Calculus is as follows:

8.3.1 Modus ponendo ponens in Predicate Calculus

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

Instantiating f as be_human and g as be_mortal has the following result:

8.3.2 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:

8.3.3 Rephrasing modus ponendo ponens in DBS

inference: α be_human implies α be_ mortal.
input: Socrates be_human.[10]
output: Socrates be_mortal.

Shown here with input for modus ponens (individual), the inference works just as well for particular (8.2.6, 1) and universal (8.2.6, 2) input.

Using the DBS data structure, the inference applies as follows to the modus ponendo ponens input of 8.3.3:

8.3.4 Applying modus ponendo ponens as formalized in DBS

	antecedent			*consequent*	
pattern level	noun: α fnc: be_human prn: K	verb: be_human arg: α prn: K	$\Rightarrow$	noun: (α K) fnc: be_mortal prn: K+1	verb: be_mortal arg: (α K) prn: K+1
		$\Uparrow$			$\Downarrow$
content level	sur: socrates noun: [person x] cat: snp sem: nm m fnc: be_human prn: 23	verb: be_human cat: #ns3′ decl sem: pres arg: [person x] prn: 23		sur: socrates noun: [person x] cat: snp sem: nm m fnc: be_mortal prn: 23+1	verb: be_mortal cat: #ns3′ decl sem: pres arg: [person x] prn: 23+1

[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 (CC 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.

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 8.3.1, and premise 2 as a singular, corresponding to $\exists$y. Therefore, the noun pattern α in 8.3.4 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.

8.4 Modus Tollendo Tollens

Modus tollendo[11] tollens serves as the individual version of 8.2.7. A standard representation in Predicate Calculus is as follows:

8.4.1 Modus tollendo tollens in Predicate Calculus

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

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

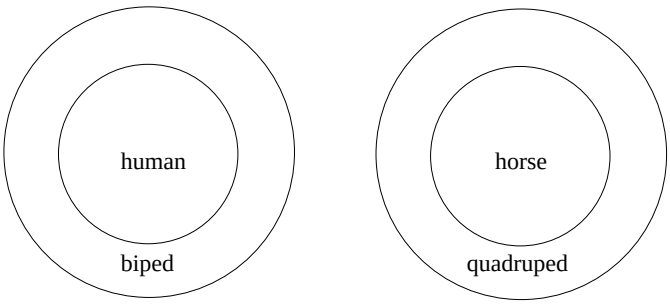
8.4.2 Instantiating modus tollendo tollens

premise 1: For all x, if x is human,[12] 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 8.4.2 may be depicted as follows:

[11] 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.

8.4.3 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.

8.4.4 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 8.2.7; as in 8.4.3, the variants differ solely in their input and output.

Let us conclude with the translation of 8.4.4 into the data structure of DBS:

8.4.5 Applying modus tollendo tollens as formalized in DBS

$$\begin{array}{ccccccc}
\textit{pattern level} & \begin{bmatrix}\text{noun: } \alpha \\ \text{fnc: be_quadr.} \\ \text{prn: K}\end{bmatrix} & \begin{bmatrix}\text{verb: be_quadr.} \\ \text{arg: } \alpha \\ \text{prn: K}\end{bmatrix} & \Rightarrow & \begin{bmatrix}\text{noun: } (\alpha\ \text{K}) \\ \text{fnc: be_human} \\ \text{prn: K+1}\end{bmatrix} & \begin{bmatrix}\text{verb: be_human} \\ \text{arg: } (\alpha\ \text{K}) \\ \text{sem: } \textit{not} \\ \text{prn: K+1}\end{bmatrix} \\
 & & \Uparrow & & & \Downarrow & \\
\textit{content level} & \begin{bmatrix}\text{sur: pegasus} \\ \text{noun: (horse x)} \\ \text{sem: nm m} \\ \text{fnc: be_quadr.} \\ \text{prn: 23}\end{bmatrix} & \begin{bmatrix}\text{verb: be_quadr.} \\ \text{sem:} \\ \text{arg: (horse x)} \\ \text{prn: 23}\end{bmatrix} & & \begin{bmatrix}\text{sur: pegasus} \\ \text{noun: (horse x)} \\ \text{sem: nm m} \\ \text{fnc: be_human} \\ \text{prn: 23+1}\end{bmatrix} & \begin{bmatrix}\text{verb: be_human} \\ \text{sem: } \textit{not} \\ \text{arg: (horse x)} \\ \text{prn: 23+1}\end{bmatrix}
\end{array}$$

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 (CC: mediated reference 3.1.3, sequential application 3.6.2).

8.5 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 (8.2.2).

8.5.1 Modus BARBARA in Predicate Calculus

premise 1: $\forall x[\mathsf{f}(x) \rightarrow \mathsf{g}(x)]$
premise 2: $\forall y[\mathsf{g}(y) \rightarrow \mathsf{h}(y)]$
conclusion: $\forall z[\mathsf{f}(z) \rightarrow \mathsf{h}(z)]$

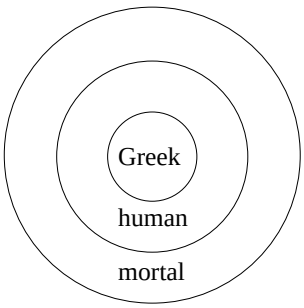
The middle term is g. If f is realized as be_Greek, g as be_human, and h as be_mortal, then the syllogism reads as follows:

8.5.2 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 8.5.2 may be depicted as follows:

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

8.5.4 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 8.5.4 into the data structure of DBS:

8.5.5 Applying modus BARBARA as formalized in DBS

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

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 (8.2.6), the reconstruction of BARBARA constitutes the universal, of DARII 8.6.5 the particular, and of modus ponendo ponens 8.3.4 the individual variant.

Next let us turn to a syllogism with the vowel **E** in its name, where **E** indicates a universal negative (8.2.2). 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:

8.5.6 Modus CELARENT in Predicate Calculus

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

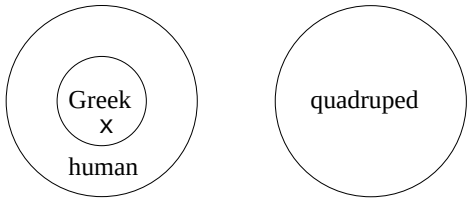
The middle term is f. If f is realized as human, g as quadruped, and h as Greek, then the syllogism reads as follows:

8.5.7 Instantiating CELARENT in Predicate Calculus

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

The set-theoretic constellation underlying modus CELARENT in 8.5.7 may be depicted as follows:

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

8.5.9 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 8.5.9 into the data structure of DBS:

8.5.10 Applying modus CELARENT as formalized in DBS

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

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 ∀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*].

8.6 Modi DARII and FERIO

The DBS variants of modus BARBARA (8.5.5) and modus CELARENT (8.5.10) have shown the treatment of the categorical judgements (8.2.2) **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:

8.6.1 Modus DARII in Predicate Calculus

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

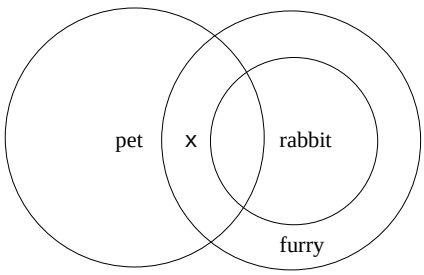
The middle term is f. If f is instantiated as be_rabbit, g as be_furry, and h as be_pet, then the syllogism reads as follows:

8.6.2 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 8.6.2 may be depicted as follows:

8.6.3 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:

8.6.4 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 8.6.4 into the data structure of DBS:

8.6.5 Applying modus DARII as formalized in DBS

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

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 (8.2.6) 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:

8.6.6 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: $\exists z[h(z) \wedge \neg g(z)]$

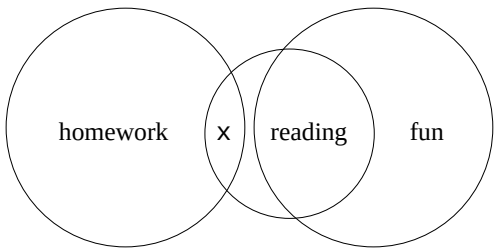
The middle term is g. If f is instantiated as is homework, g as is fun, and h as is reading, then the syllogism reads as follows:

8.6.7 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 8.6.7[13] may be depicted as follows:

8.6.8 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.

Consider the DBS inference schema for the set-theoretic constellation:

8.6.9 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 8.6.9 into the data structure of DBS:

[13] Instantiating FERIO with f = dog, g = bird and h = animal also satisfies 8.6.7.

8.6.10 Applying modus FERIO as formalized in DBS

pattern level	noun: α fnc: be_homew. prn: K	verb: be_homew. arg: α prn: K	$\Rightarrow$	noun: (α K) fnc: be_fun prn: K+1	verb: be_fun sem: *not* arg: (α K) prn: K+1	
		$\Uparrow$			$\Downarrow$	
content level	noun: reading cat: snp sem: pl sel fnc: be_homew. prn: 23	verb: be_homew. arg: reading prn: 23		noun: (reading 23) cat: snp sem: pl sel fnc: be_fun prn: 23+1	verb: be_fun sem: *not* arg: (reading 23) prn: 23+1	

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: *not*] 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 (8.2.7) in the variant *particular*.

8.7 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 (8.2.2).

8.7.1 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:

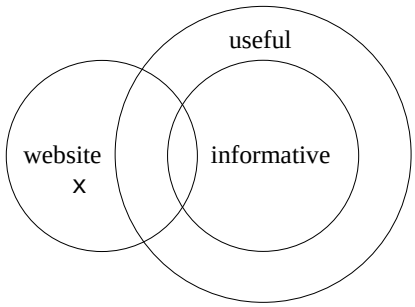
8.7.2 Instantiating Modus BAROCO

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 8.7.2 may be depicted as follows:

8.7.3 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:

8.7.4 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 8.7.4 into the data structure of DBS:

8.7.5 BAROCO in DBS

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

The reconstruction of BAROCO in DBS joins inferences of the kind negated antecedent and negated consequent (8.2.9) in the variant *particular*. It differs from DARII (8.6.5) 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.

8.7.6 Modus BOCARDO in Predicate Calculus

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

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:

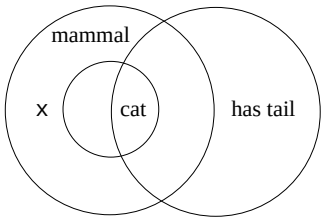
8.7.7 Instantiating modus BOCARDO

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).[14]

The set-theoretic constellation underlying modus BOCARDO in 8.7.7 may be depicted as follows:

8.7.8 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.

8.7.9 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 8.7.8 into the data structure of DBS:

8.7.10 Applying BOCARDO as a DBS inference

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

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 (8.6.9), which is unrestricted and thus compatible with universal, particular, and individual input (compatibility by omission)

8.8 Combining S- and C-Inferencing

Functional equivalence (CC Sects. 1.1, 15.1) 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 8.1.2 and CC 5.1.4, continues with a lexical S-inference coding a hypernymy (op. cit. 5.2.2), and concludes with another C-inference:

8.8.1 Mixing S- and C-inference in a train of thought

1 C-inference: α is hungry $\Rightarrow$ α is cranky (8.1.2)
$\Uparrow$ $\Downarrow$
input: Laura is hungry output: Laura is cranky

2 C-inference: α is cranky $\Rightarrow$ α needs food
$\Uparrow$ $\Downarrow$
input: Laura is cranky output: Laura needs food

3 S-inference: Laura eats β $\Rightarrow$ $\beta\ \varepsilon$ {apple, banana, cookie,. . . , strawberry}
$\Uparrow$ $\Downarrow$
input: Laura eats food output: Laura eats apple or cookie or banana, ..., or strawberry ...

[14] 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 C-inference: α eats cookie $\Rightarrow$ α is agreeable again
$\Uparrow$ $\Downarrow$
input: Laura eats cookie output: Laura is agreeable again

The S-inference 3 illustrates a lexical alternative to the syllogisms analyzed in Sects. 8.3–8.7, namely a hypernymy, which is defined as follows.

8.8.2 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, ...}
...

The set-theoretic structure of a hypernymy[15] 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[16] corpus).

8.9 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.

8.9.1 Common sense reasoning based on analogy

1 C-inference: α has no fuel $\Rightarrow$ α does not start
$\Uparrow$ $\Downarrow$
input: truck has no fuel output: truck does not start

[15] For the corresponding hyponymy see CC 9.1.1.
[16] Reference-Monitor corpus structured into Domains (CLaTR Sect 15.3).

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 8.9.1, 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 8.8.2, while the C-inference 4 derives the desired result.

Using the data structure of DBS, the data-driven application of the second inference in 8.9.1 may be shown as follows:

8.9.2 APPLYING THE C-INFERENCE 2 OF 8.9.1

pattern level

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

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

$\Uparrow$ $\Downarrow$

content level

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

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

8.10 Conclusion

The valid syllogisms and the corresponding DBS S-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 (substitution-driven sign-based), 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 (data-driven agent-based).

Bibliography

Aristotle (1974). *The Complete Works of Aristotle*, edited by Jonathan Barnes, 2 vols., Princeton University Press

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

Euler, L. (1761) *Lettres a Une Princesse d'Allemagne*, Vol. 2: Letters No. 102–108. St. Petersburg: Imperial Academy of Sciences, 1768–1772

Frege, G. (1879) *Begriffsschrift. Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle: L. Nebert

NLC2 = Hausser R. (2017) *A Computational Model of Natural Language Communication – Interpretation, Inferencing, and Production in Database Semantics*, available at *lagrammar.net*. 1st edition, (2006), Springer, pp. 360

Read, S. (2017) "Aristotle's Theory of the Assertoric Syllogism," `https://www.st-andrews.ac.uk/~slr/The_Syllogism.pdf`

TExer3 = Hausser, R. (2020) *Twentyfour Exercises in Linguistic Analysis – DBS Software Design for the Hear and the Speak Mode of a Talking Robot*, pp. i–vi, 1–324, DOI: 10.13140/RG.2.2.36392.01282, `lagrammar.net`

Venn, J. (1881) *Symbolic Logic*, London: MacMillan

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

Topics of this small volume are a content-addressable database schema for storage and retrieval; a computational data structure for processing with the time-linear algorithm of LA-grammar; low computational complexity (linear); an iterative rather than recursive approach to repetition in grammar; a computational reconstruction of the classical Semantic kinds concept, indexical, and name with their respective computational Mechanisms of type-token matching, pointing at values of the on-board orientation system, and baptism by inserting the named referent as the core value into a lexical name proplet; the transfer channel from speaker to hearer in communication; and reference as a purely cognitive mechanism restricted to nouns.

Drawing on earlier work, the basic idea is a data-driven agent-based ontology suitable for building a talking robot, in contradistinction to the substitution-driven sign-based alternative. An agent-based ontology includes the processing media and the sensory media, and their dual modalities in recognition and action. It also includes the component structure and functional flow of a cognitive agent with an interface component, a memory with a now front for current processing, and an operation component for the hear, the think, and the speak mode.