

THEORETICAL ESSAY

DATABASE SEMANTICS

Roland HAUSSER  

Universität Erlangen-Nürnberg (FAU)

ABSTRACT

For long-term upscaling, the computational reconstruction of a complex natural mechanism must be *input-output equivalent* with the prototype, i.e. the reconstruction must take the same input and produce the same output in the same processing order as the original. Accordingly, the modeling of natural language communication in Database Semantics (DBS) uses a time-linear derivation order for the speaker's output and the hearer's input. The language-dependent surfaces serving as the vehicle of content transfer from speaker to hearer are raw data without meaning or any grammatical properties whatsoever, but measurable by natural science.

RESUMO

Para garantir um aprimoramento de longo prazo, a reconstrução computacional de um mecanismo natural complexo deve ser *equivalente* com o protótipo em *termos de entrada e saída*, ou seja, a reconstrução deve ter a mesma entrada e produzir a mesma saída na mesma ordem de processamento que o original. Seguindo este princípio, a modelagem da comunicação de linguagem natural da Database Semantics (DBS) usa uma ordem de derivação temporalmente linear para a saída do falante e a entrada do ouvinte. As superfícies dependentes da linguagem que servem como veículo de transferência de conteúdo do falante para o ouvinte são dados brutos sem significado ou quaisquer propriedades gramaticais, mas mensuráveis pelas ciências naturais.

KEYWORDS

Agent-Based; Interface Component; On-Board Database and Orientation System; Recognition and Action; Speak, Think, and Hear

OPEN ACCESS

EDITORS

- Miguel Oliveira, Jr. (UFAL)
- René Almeida (UFS)

REVIEWERS

- Alexandre Rademaker (IBM)
- Raquel Freitag (UFS)

DATES

- Received: 05/03/2021
- Accepted: 07/10/2021
- Published: 12/14/2021

HOW TO CITE

HAUSSER, Roland (2021). Database Semantics. *Cadernos de Linguística*, v. 2, n. 1, e382.

Mode; Concept, Indexical, and Name; Type-Token Matching, Pointing, and Baptism; Raw Data for Surface Transfer.

PALAVRAS-CHAVE

Baseado em Agente; Componente de Interface; Banco de Dados On-Board e Sistema de Orientação; Reconhecimento e Ação; Modo Falar, Pensar e Ouvir; Conceito, Indexicalidade e Nome; Correspondência de Tipo-Token, Apontar e Batismo; Dados Brutos para Transferência de Superfície.

PART I: FOUNDATION

To start from familiar ground, let us begin with Predicate Calculus as a widely accepted representation of meaning/content, followed by the DBS alternative. The most basic difference between them is their ontology: Predicate Calculus is sign-based substitution-driven, while DBS is agent-based data-driven.

BRIEF OUTLINE OF PREDICATE CALCULUS

In Predicate Calculus (PredC), elementary meanings are treated as mini-propositions which denote truth values relative to a set-theoretic model. Defined as functors which may differ in the number of arguments, e.g. $f(x)$ vs. $f(x,y)$, they are connected by the operators of Propositional Calculus (PropC) and the quantifiers of PredC:

1.1. NOUN, VERB, AND ADJ FLATTENED INTO MINI-PROPOSITIONS

noun **dog**: interpreted as “x is a dog” and written as “dog'(x)”
adj **black**: interpreted as “x is black” and written as “black'(x)”
verb 1-place **snore**: interpreted as “x snores” and written as “snore'(x)”
verb 2-place **find**: interpreted as “x finds y” and written as “find'(x, y)”
verb 3-place **give**: interpreted as “x gives y z” and written as “give'(x, y, z)”

This allows to represent, for example, The dog found a bone as three mini-propositions which are coordinated with the propositional operator \wedge , and have the variables x and y bound by two \exists quantifiers:

1.2. PREDC REPRESENTATION OF *The dog found a bone*

$$\exists x \exists y [\text{dog}'(x) \wedge \text{bone}'(y) \wedge \text{find}'(x, y)]$$

The different meanings of dog, find, and bone are defined by the denotation function F and the assignment function g of a formal model:

1.3. MINIMAL MODEL FOR THE PREDC FORMULA 1.2

Let M be a model $\langle A, B, F, g \rangle$, where A is an infinite set of objects or individuals, B a finite set of basic expressions, F a denotation function from B into A^* , and g an assignment function from variables into A^* .

For illustration, let us define A, B, F, and g as follows:

$A = \{a_1, a_2, a_3\}$

$B = \{\text{dog, find, bone}\}$

$F(\text{dog}) = \{a_1\}$

$F(\text{find}) = \{<a_1 a_2>\}$

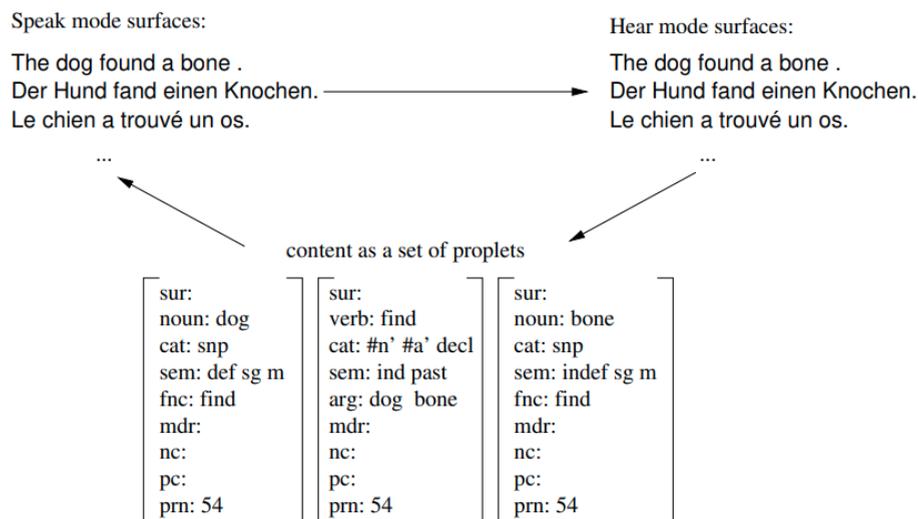
$F(\text{bone}) = \{a_2\}$

$g(x) = a_1$

$g(y) = a_2$

Based on these definitions, the formula 1.2 is well-formed and true in M . However, if we defined $F(\text{dog})$ as a_3 , for example, the formula would be false.

2. CONTENT IN DBS: AGENT-BASED DATA-DRIVEN



This content is defined as a set (order-free) of three proplets, defined as non-recursive feature structures with ordered attributes, connected by semantic relations of structure coded by address and a shared prn (proposition number) value, here 54. Proplets are the computational data structure of DBS.

The speak mode takes a content as input and produces a surface as output. The hear mode takes a surface as input and produces a content as output. As a minimal standard for natural language communication to be successful, (i) a content mapped by the speak mode into a surface and (ii) the content resulting from the hear mode's interpretation of that surface must be the same (15.1, (ii)).

3. ADDITIONAL COMMUNICATION CONDITIONS

For successful communication in the wider sense, the hearer must infer the speaker's intent by reconstructing a possible inference by the speaker for a nonliteral content production, resulting in a nonliteral meaning (pragmatics).

3.1. FIRST PRINCIPLE OF PRAGMATICS (POP-1, FOCL 4.3.3)

The speaker's utterance meaning₂ is the use of the sign's literal meaning₁ relative to an (agent-)internal context (of use).

A content is related to both meaning₁ and meaning₂ as follows:

3.2. DEFINING CONTENT IN TERMS LANGUAGE

A content is a set of connected proplets without sur values.

A meaning₁ is related to a content as follows:

3.3. DEFINING LANGUAGE MEANING₁ IN TERMS OF CONTENT

A meaning₁ is a content with language-dependent sur values.

In DBS, language and nonlanguage cognition use the same kind of proplets connected by the same semantic relations of structure. The proplets of language and nonlanguage cognition differ in the presence vs. absence of language-dependent sur values (2.1).

4. SEMANTICS, SYNTAX, CONTENT, AND MECHANISMS

Linguistics, symbolic logic, and philosophy use the following notions:

4.1. RELATED NOTIONS IN LINGUISTICS, LOGIC, AND PHILOSOPHY

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

These are not merely different terms for the same things, but different terms for different aspects of the same things. In DBS, the distinctions are related as follows:

4.2. 1ST CORRELATION: SEMANTICS AND THEIR SYNTAX

<p>(i) <i>Semantic kind</i></p> <ol style="list-style-type: none"> 1. referent 2. property 3. relation 	<p>(ii) <i>Syntactic kind</i></p> <p>noun</p> <p>adn, adv, adnv, intransitive verb</p> <p>transitive verb</p>
---	---

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

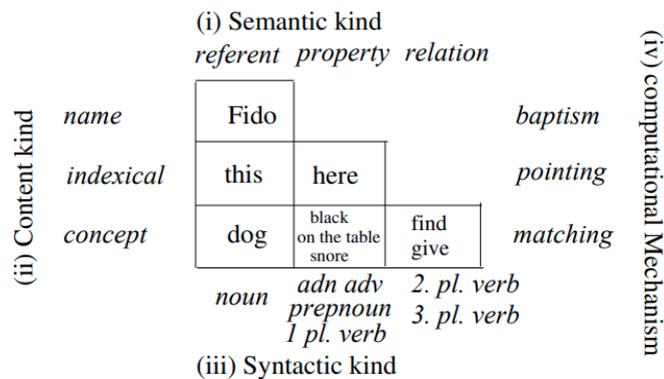
4.3 2ND CORRELATION: CONTENTS AND THEIR MECHANISMS

<p>(iii) <i>Content kind</i></p> <ol style="list-style-type: none"> 4. name 5. indexical 6. concept 	<p>(iv) <i>computational Mechanism</i></p> <p>baptism</p> <p>pointing</p> <p>matching</p>
--	---

The two dichotomies provide 12 basic notions, six basic pairs, and two correlations.

Theoretically, there are 12 classes of proplets which differ in the Semantic, Syntactic, Content, and Mechanism kind. Empirically, however, there are only six. They constitute the semantic building blocks of DBS cognition in general and natural language communication in particular. The six classes of proplets form the *cognitive square* of DBS, which is considered universal:

4.4. COGNITIVE SQUARE OF DBS



The most general content kind is the concepts which occur as all three semantic kinds *referent*, *property*, and *relation*. The most general semantic kind is the referents, which occur as *name*, *indexical*, and *concept*.

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

5. CONTENT KINDS AND COMPUTATIONAL MECHANISMS

Of the content kinds *concept*, *indexical*, and *name*, it is the concepts which establish the interaction between the agent's cognition and the cognition-external raw data.

5.1. CONCEPTS OCCUR IN ALL THREE SEMANTIC KINDS

- (i) *referent*, e.g. **dog**
- (ii) *property*, e.g. **black, on the table, snore**
- (iii) *relation*, e.g. **find, give**.

The computational Mechanism of concepts is type/token¹ matching (6.1, 6.3, 7.1, 7.3). The three Content kinds of the Semantic kind *referent*, i.e. names, indexicals, and concepts, all have literal use, but only the concepts allow figurative use.

The computational Mechanism of indexicals is pointing at values of the agent's onboard orientation system (OBOS).

5.2. INDEXICALS OCCUR IN TWO OF THE THREE SEMANTIC KINDS

- (i) *referent*, e.g. **pro1, pro2, pro3, and**
- (ii) *property*, e.g. **here, now**

The semantic kind absent in indexicals is *relation*. Also, indexicals have no non-literal use. Indexicals depend on the pattern matching mechanism of concepts insofar as concepts are the *range* of the indexicals' pointing mechanism. For example, if the indexical *here* points at the [S: veranda] feature of the agent's OBOS, then the interpretation of the indexical depends on the concept 'veranda'.

The functioning of names is a controversial topic in philosophy (Frege 1892, Russell 1905, Kripke 1972), but baptism is at the center. In DBS, the computational Mechanism of names inserts the address of a *named referent* into lexical name proplets as their core value (CASM 2017):

¹ C.S. Peirce (1906, CP Vol.4, p. 375). The type/token distinction goes back to Aristotle's distinction between the *necessary* and the *accidental* (Metaphysics, Books ζ and η).

5.3 BAPTISM PROVIDING REFERENT TO LEXICAL NAME PROPLET

<p>sur: Peter noun: [person x] cat: snp sem: sg m fnc: mdr: nc: pc: prn</p>	←	<i>named referent</i>
---	---	-----------------------

The semantic kinds absent in names are *property* and *relation*. Like indexicals, names have no nonliteral use.

5.4 NAMES OCCUR IN ONLY ONE OF THE THREE SEMANTIC KINDS

(i) *referent*, e.g. **Peter**

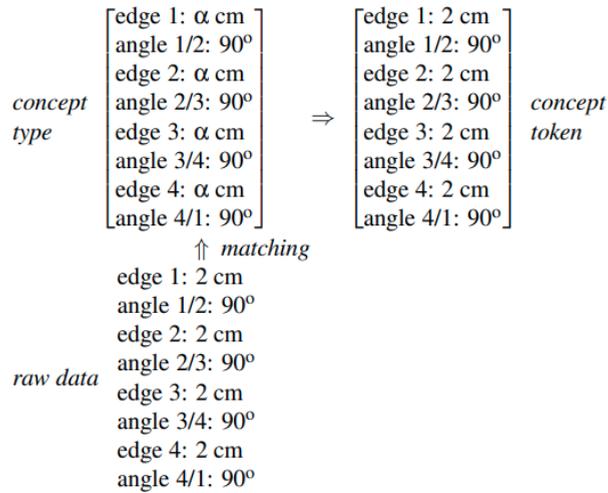
Names depend on the pattern matching mechanism of concepts insofar as the named referent, e.g. [person x], is an address, which consists in part of a concept.

6. USING CONCEPTS OF GEOMETRY IN RECOGNITION AND ACTION

Natural languages may differ substantially, as in being isolating, inflectional, agglutinating, accusative, ergative, etc. Underneath, however, there is the universal hardware level which all speaker-hearers have in common. In accordance with Philip Lieberman (1984, 2000), DBS locates Chomsky's intangible "language ability" in the human capability to produce and recognize raw acoustic data as language-dependent surfaces in the medium (CC 11.2.1) of speech. In evolution, the medium of writing, including Braille, was added later.

The functioning of a concept in nonlanguage recognition may be shown as follows:

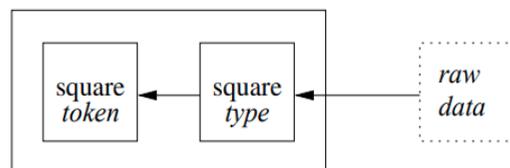
6.1. RECOGNITION: RAW DATA MATCHING TYPE RESULT IN TOKEN



The type defines the concept of a *square*. Replacing its variables with constants provided by the raw data, here α with *2cm*, results in a concept token.

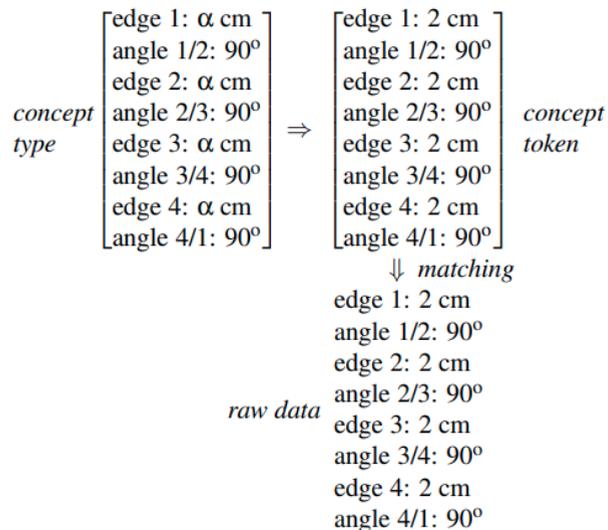
More schematically, the agent's nonlanguage recognition of a square by means of type-token matching may be shown as follows:

6.2. USING A CONCEPT TYPE IN NONLANGUAGE RECOGNITION



As the action counterpart to 6.1, consider the adaptation of a concept type into a concept token by the agent's cognition for a purpose:

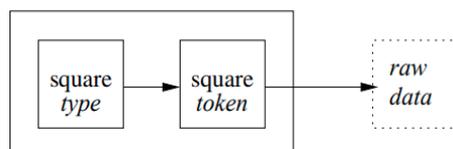
6.3. ACTION: TYPE-TOKEN ADAPTATION RESULTS IN RAW DATA



Adapting the type to a purpose results in a token which is realized as raw data.

More schematically, the agent's nonlanguage action of drawing a square by means of a type-token adaptation may be shown as follows:

6.4. USING A CONCEPT TOKEN IN NONLANGUAGE ACTION



In DBS, the definition of concept types, corresponding concept tokens, and raw data relies on the natural sciences, here geometry. The raw input data are provided by the agent's interface component. They are recognized as an instance of the twodimensional shape square because there are four lines of equal length and the angle of their intersections is 90°.

The concept *square* may be extended to other two-dimensional geometric objects:

6.5. SIMILARITY AND DIFFERENCE BETWEEN CONCEPT SHAPE TYPES

[place holder: equilateral triangle sensory modality: vision semantic field: two-dim geom. content kind: concept shape: [edge 1: α cm angle 1/2: 60° edge 2: α cm angle 2/3: 60° edge 3: α cm angle 3/4: 60° samples: a, b, c,...]	[place holder: rectangle sensory modality: vision semantic field: two-dim geom. content kind: concept shape: [edge 1: α cm angle 1/2: 90° edge 2: β cm angle 2/3: 90° edge 3: α cm angle 3/4: 90° edge 4: β cm angle 4/1: 90° samples: a', b', c',...]	[place holder: square sensory modality: vision semantic field: two-dim geom. content kind: concept shape: [edge 1: α cm angle 1/2: 90° edge 2: α cm angle 2/3: 90° edge 3: α cm angle 3/4: 90° edge 4: α cm angle 4/1: 90° samples: a'', b'', c'',...]
--	--	--

For retrieving the correct type, i.e. the one best matching the raw data at hand, the examples expand the concept analysis by embedding the geometric type shape into non-recursive feature structures with ordered attributes which specify the sensory modality, the semantic field, and whatever else is useful to aid retrieval of the type most suitable for matching the raw data at hand.

On the one hand, the lines and angles of two-dimensional geometry have counterparts in neurology, such as the line, edge, and angle detectors in the optical cortex of the cat (Hubel and Wiesel 1962), and the iconic or sensory memories from which the internal image representations are built (Sperling 1960) and Neisser (1967). On the other hand, robotic vision (Wiriathamabhum et al. 2016) applies the natural science of optics in ways which differs markedly from the natural prototype (Pylyshyn 2009).

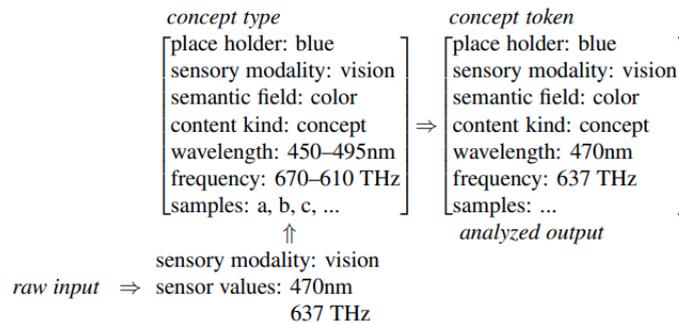
This is analogous to the difference between the natural flight of (i) birds, bats, and butterflies (flapping wings), and the artificial flight of (ii) air planes (fixed wings), and (iii) helicopters (rotors), all of which satisfy the laws of aero-dynamics (CLaTR Sect. 1.1). The list goes on with differences in earthbound locomotion (legs vs. wheels), and power supply (digestion vs. electricity).

Input-output equivalence with the natural prototype and alternative uses of the natural sciences are not in conflict. As illustrated in 8.3, input-output equivalence affects macro-processing while alternative uses of the natural sciences affect micro-processing (CLaTR Sect. 1.1).

7. USING CONCEPTS OF COLOR IN RECOGNITION AND ACTION

Another homogeneous class of concepts are the colors. Their basic principles of recognition and production resemble those of the geometric shapes as follows:

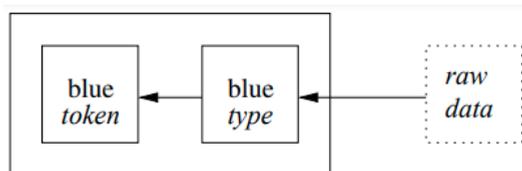
7.1. TYPE MATCHING RAW DATA IN COLOR RECOGNITION



The analyzed output token results from replacing the wavelength and frequency intervals of the type with the measurements of the raw input data.

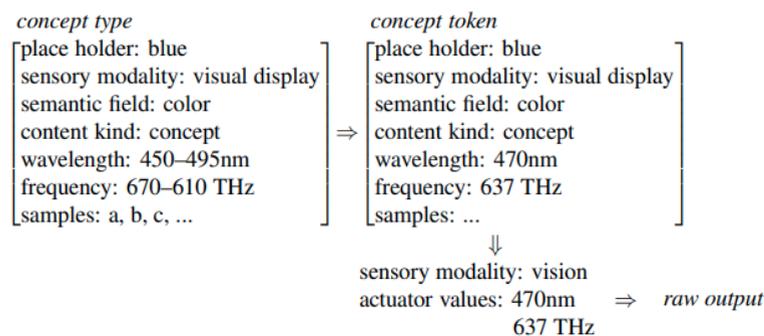
More schematically, the agent’s nonlanguage recognition of colors by means of typetoken matching may be shown as follows:

7.2. DIRECT USE OF A CONCEPT IN NONLANGUAGE RECOGNITION



In the corresponding nonlanguage action, the type is adapted to a purpose, as in the cuttlefish *Metasepia pfefferi* turning on the color blue:

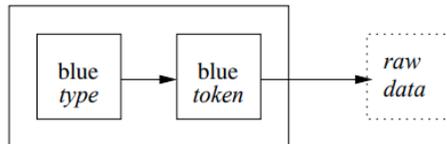
7.3. TYPE-TOKEN ADAPTATION IN COLOR PRODUCTION (CC 11.3.4)



Cognition adapts the type to a purpose by replacing the wavelength interval of 450–495nm and frequency interval of 670–610 THz of the type with the specific values of 470nm and 637 THz, resulting in a token. In the cuttlefish, these values are realized by natural actuators for color control (chromatophores) as raw data.

More schematically, the agent's nonlanguage action of colors by means of type-token adaptation may be shown as follows:

7.4. DIRECT USE OF A CONCEPT IN NONLANGUAGE ACTION



What has been shown for the color concept blue may be extended to the whole class:

7.5. SIMILARITY AND DIFFERENCE BETWEEN COLOR CONCEPT TYPES

[place holder: red sensory modality: vision semantic field: color content kind: concept wavelength: 700-635 nm frequency: 430-480 THz samples: a, b, c, ...]	[place holder: green sensory modality: vision semantic field: color content kind: concept wavelength:495-570 nm frequency: 526-606 THz samples: a', b', c', ...]	[place holder: blue sensory modality: vision semantic field: color content kind: concept wavelength: 490-450 nm frequency: 610-670 THz samples: a'', b'', c'', ...]
---	---	--

The three types differ in their wavelength and frequency intervals, and their place holder and samples values; they share the sensory modality, semantic field, and content kind values. For convenience, the place holders are in English.

In summary, the use of concepts in recognition vs. action differs in the origin of the specialization from a concept type to a concept token. In recognition, the specialization of the type into a token happens in the pattern matching with raw data. In action, the type is specialized into a token by the agent's cognition as an adaptation to a purpose for use of the token to produce of suitable raw data.

8. COMBINING CONCEPTS INTO NON-LANGUAGE CONTENT

The interaction of nonlanguage and language recognition and action is required for autonomous *reference*. For instance, requesting a robot to pick a blue square from a collection of colored two-dimensional geometric objects requires the artificial agent to understand the request, to recognize the object in question as the modifier|modified blue | square (reference), and to perform the action (hand-eye coordination).

The following example shows the (a) concept type blue, the (b) associated nonlanguage proplet, and (c) the language proplet for German:

8.1. EMBEDDING THE CONCEPT BLUE INTO ADJ PROPLETS

a. *concept*

```
[place holder: blue
sensory modality: vision
semantic field: color
content kind: concept
wavelength: 490-450 nm
frequency: 610-670 THz
samples: a'', b'', c'', ... ]
```

b. *nonlanguage proplet*

```
[place holder: blue
sensory modality: vision
semantic field: color
adj: content kind: concept
wavelength: 490-450 nm
frequency: 610-670 THz
samples: a'', b'', c'', ... ]
cat: adn
sem: pad
...
prn:
```

c. *language proplet*

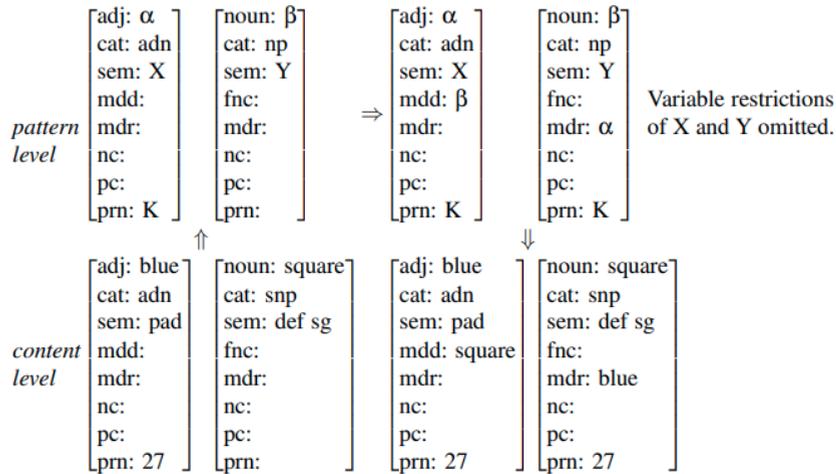
```
[sur: blaus
[place holder: blue
sensory modality: vision
semantic field: color
adj: content kind: concept
wavelength: 490-450 nm
frequency: 610-670 THz
samples: a'', b'', c'', ... ]
cat: adn
sem: pad
...
prn:
```

Concept (a) is common to the proplets (b) and (c), serving as their respective core value. The difference between the non-language proplet (b) and language proplet (c) is the absence vs. presence of a sur feature. The concept of all three is a type because the wavelength and frequency values are intervals instead of constants.

In evolution, nonlanguage cognition precedes language cognition and supplies the latter with essential constructs such as the media, sensory modalities, content kinds, and mechanisms for the combination of concept tokens into complex nonlanguage content. Accordingly, DBS language cognition uses the same semantic kinds (referent, property, relation), the same content kinds (concept, indexical, name), the same computational mechanisms (type-token matching, pointing, baptism), and the same syntactic kinds (4.4) as DBS non-language cognition.

Using the same elements and the same mode of composition does not mean, however, that nonlanguage and language cognition must be identical. For example, it is not a foregone conclusion that the nonlanguage composition corresponding to English the+blue+square should code the contribution of the definite article as a separate nonlanguage proplet. It is just as possible that nonlanguage cognition treats definiteness in the sense of *known* or *familiar* as a property of square instead of a separate determiner, as in the following hypothetical operation:

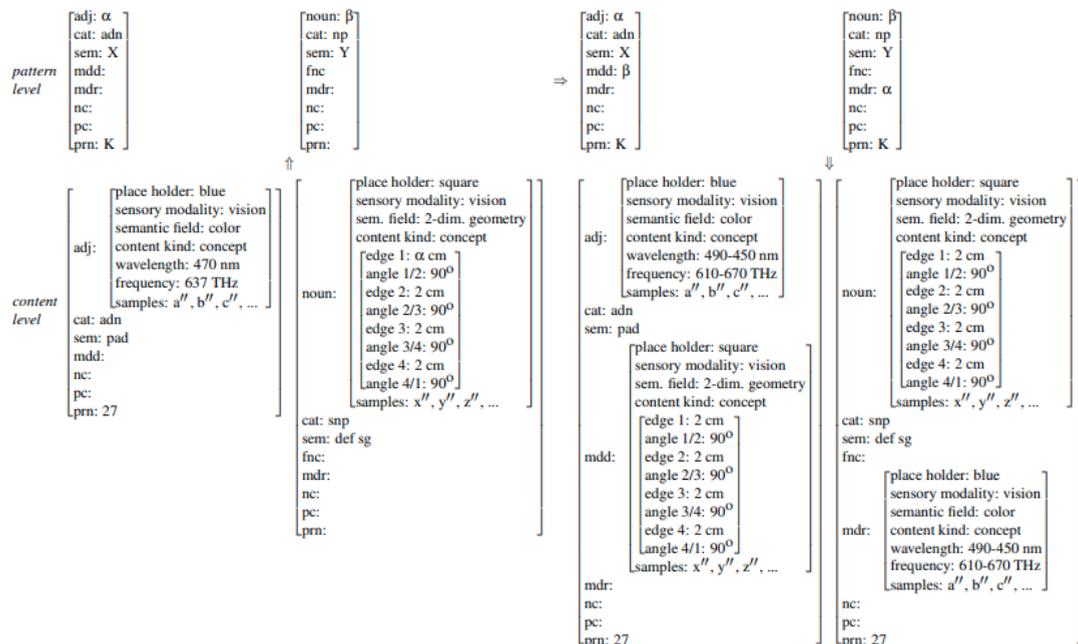
8.2. NONLANGUAGE MODIFIER | MODIFIED OPERATION



Here the value def originates in the noun rather than the determiner.

The corresponding operation with the place holders replaced by the explicit nonlanguage concepts blue and square at the content level is shown as follows:

8.3 NONLANGUAGE CONCATENATION WITH EXPLICIT CONCEPTS

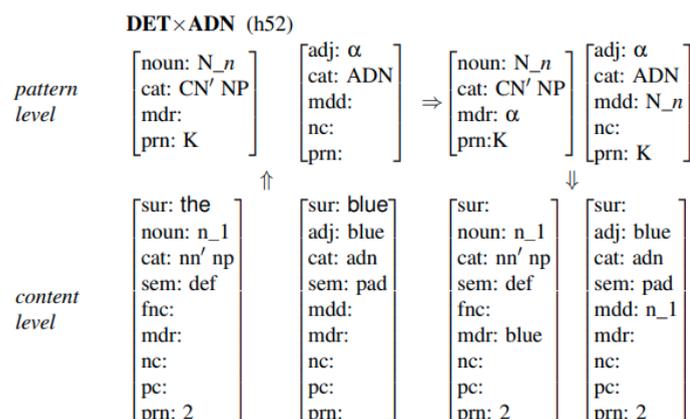


By binding the explicitly defined concepts of the input content to the variables α and β serving as core values in the input pattern, the output pattern produces a result with explicitly defined values at the content level.

9. HEAR MODE OPERATIONS WITH PLACE HOLDERS

The equivalent English language composition of the blue square differs from the nonlanguage composition 8.3 in that the [sem: def sg] feature of the square proplet originates in a separate language proplet,² the, such that three rather than two proplets require concatenation in the hear mode and production in the speak mode.

9.1. CROSS-COPYING THE AND BLUE WITH DET×AND

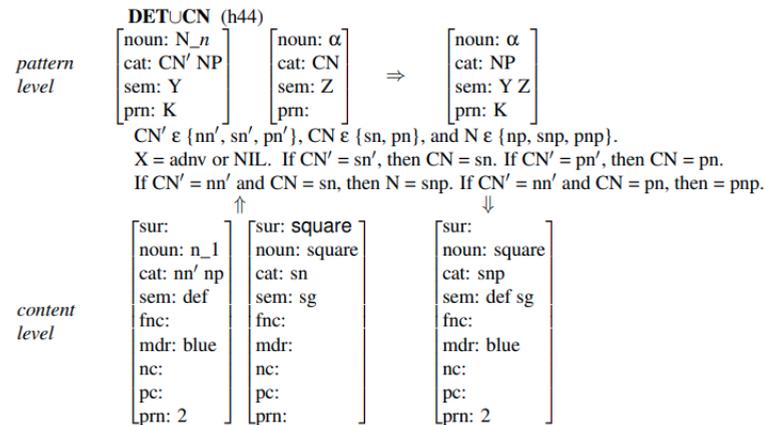


The core value of the determiner the matching the first input pattern is the substitution variable n_1 . Cross-copying establishes the modifier|modified relation by writing blue into the mdr slot of *the* and n_1 into the mdd slot of *blue* in the output. The operation number h52 refers to the DBS hear mode grammar in TExer 6.3.1.

The next proplet provided by automatic word form recognition is *square*. The proplet matches the second input pattern of the hear mode operation DETuCN.

² “Analytization” (Haspelmath&Michaelis 2017).

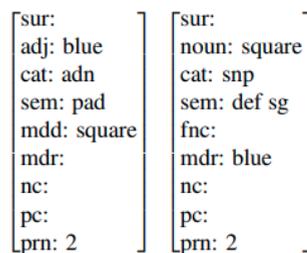
9.2 ABSORBING SQUARE INTO THE WITH DET_UCN



The connective U indicates *absorption* by global substitution: the determiner proplet absorbs the common noun proplet by replacing all instances of the variable n₁ with the value square, after which the common noun proplet is discarded.

The time-linear hear mode concatenation of three lexical input proplets required two operations and resulted in two output proplets:

9.3. RESULT OF THE HEAR MODE OPERATIONS 9.1 AND 9.2



This equals the output of nonlanguage 8.2, except for the sur attributes.

10. SPEAK MODE OPERATIONS WITH PLACE HOLDERS

The input to the speak mode is a content, defined as a set of non-language proplets connected with the semantic relations of structure, defined by address. Consider the nonlanguage content of Lucy found a big blue square:

10.1. NON-LANGUAGE CONTENT

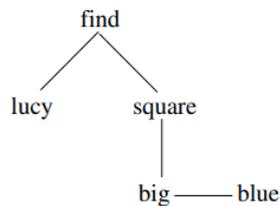
referent (noun)	relation (transitive verb)	property (adj)	property (adj)	referent (noun)
[sur: lucy noun: [person x] cat: snp sem: nm f fnc: find mdr: nc: pc: prm: 23]	[sur: ³ verb: find cat: #n' #a' decl sem: ind past arg: [person x] square mdr: nc: pc: prm: 23]	[sur: adj: big cat: adn sem: pad mdd: square mdr: nc: blue pc: prm: 23]	[sur: adj: blue cat: adn sem: pad mdd: mdr: nc: pc: big prm: 23]	[sur: noun: square cat: snp sem: def sg fnc: find mdr: big nc: pc: prm: 23]

The two-place relation *find* combines the referents [*person x*] and *square* into the proposition [*person(x)*] *find square*. The property (modifier) *big* combines with *square* by functor-argument and with *blue* by coordination. The concepts *person*, *find*, *big*, *blue*, and *square* are shown by place holder values.

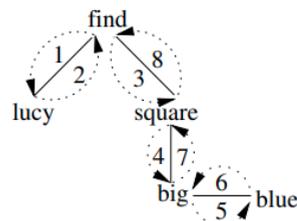
The equivalent graphical representation is shown in four views:

10.2. SEMANTIC RELATIONS UNDERLYING SPEAK MODE

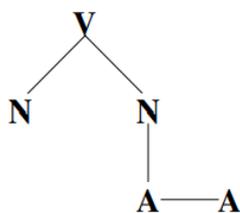
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



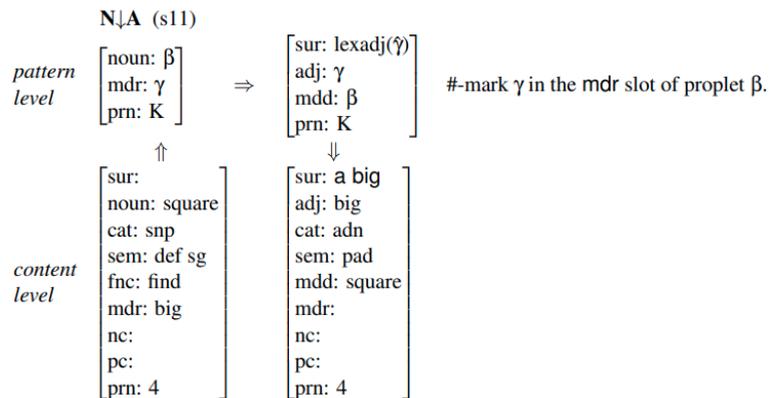
(iv) surface realization

1	2	3	4	5	6	7	8
Lucy	found	a	big	blue	square	.	
V/N	N/V	V\N	N A	A-A	A-A	A N	N V

View (i) is called the semantic relations graph (SRG). It is based on the proplets of the content 10.1 and uses the core *values* *lucy*, *find*, *square*, *big* and *blue* as nodes. View (ii) is called the signature and uses the core *attributes* N for noun, V for verb, and A for adj as nodes. View (iii) is called the *numbered arcs graph* (NAG) and supplements the SRG with numbered arcs which are used in the (iv) *surface realization*; it consists of three parallel lines which show the navigation as it activates content in the think mode and optionally realizes the language-dependent surfaces in a speak mode which rides piggyback on the think mode navigation.

Assuming the content 10.1 as input to the agent's speak mode, the intermediate speak mode navigation step in arc 4 proceeds from the noun proplet *square* to the adnominal modifier *big* with N↓A:

10.3. NAVIGATING WITH N↓A FROM SQUARE TO BIG (ARC 4)



The traversal sequence $\forall \setminus N - N \downarrow A$ complies with the Continuity Condition of DBS (NLC 3.6.5), according to which a think-speak mode operation AcB ('c' for connective) may only be followed by an operation BcC which accepts the output of AcB as input. The Continuity Condition is the think-speak mode counterpart to the timeliness in the hear mode. In fact, the Continuity Condition of the speak mode is the source of the hear mode's time-linear derivation order.

11. UNIVERSAL: AMBIGUITY AND PARAPHRASE ARE MUTUALLY EXCLUSIVE

The respective uni-directional derivations of the speak and the hear mode go naturally with a universal asymmetry between ambiguity and paraphrase:

11.1. DEFINITION OF AMBIGUITY AND PARAPHRASE

ambiguity (hear mode: several contents for the same surface)

paraphrase (speak mode: several surfaces for the same content).

It follows that ambiguity and paraphrase in natural language are mutually exclusive. Understandably, this has been overlooked³ by the sign-based substitution-driven approaches in linguistics, which analyze language without separate derivations for the speak and the hear mode, such as Phrase Structure Grammar including ST, EST, REST, GB, GPSG, HPSG, etc., and Categorical Grammar in its various guises.

12. LEXICAL VS. STRUCTURAL AMBIGUITY

A standard distinction in philology is between lexical vs. structural ambiguity. A wellknown lexical ambiguity in English is Flying airplanes can be dangerous. One reading interprets flying as a present participle and is paraphrased as to fly airplanes. The other reading interprets flying as an adnominal modifier and is paraphrased as airplanes which fly. In DBS, the two readings are distinguished as follows:

12.1. LEXICAL AMBIGUITY OF FLYING AIRPLANES

present participle

[sur: flying verb: fly cat: a' be sem: prog arg: Ø airplane mdr: mdd: nc: pc: prn: 27	[sur: airplanes noun: airplane cat: pnp sem: indef pl fnc: fly mdr: mdd: nc: pc: prn: 27
--	---

adnominal modifier

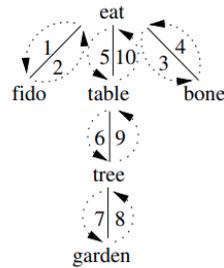
[sur: flying verb: fly cat: adj sem: pad arg: mdr: mdd: airplane nc: pc: prn: 28	[sur: airplanes noun: airplane cat: pnp sem: indef pl fnc: mdr: fly mdd: nc: pc: prn: 28
---	---

In comparison, the structural ambiguity 'Fido ate the bone on the table' has an adverbial and an adnominal reading. On the adverbial reading, it is the eating which is on the table (TExer 3.2.2). Accordingly, the modifier|modified column is attached to the predicate:

³ A notabel exception is Neumann 1998.

12.2. STRUCTURAL AMBIGUITY: ADVERBIAL (READING 1)

(iii) NAG (numbered arcs graph)



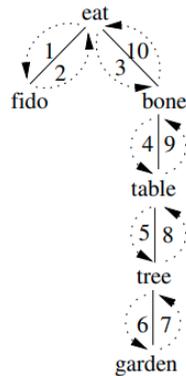
(iv) surface realization

1	2	3	4	5	6	7	8	9	10	
Fido	ate	the_bone	on_the_table	under_the_tree	in_the_garden					•
V/N	N/V	V\N	N\N	V\N	N\N	N\N	N\N	N\N	N\N	N\N

On the adnominal reading, in contrast, it is the bone which is on the table (TExer 3.2.1). Accordingly, the modifier|modified column is attached to the grammatical object:

12.3. STRUCTURAL AMBIGUITY: ADNOMINAL (READING 2)

(iii) NAG (numbered arcs graph)



(iv) surface realization

1	2	3	4	5	6	7	8	9	10	
Fido	ate	the_bone	on_the_table	under_the_tree	in_the_garden					•
V/N	N/V	V\N	N\N	N\N	N\N	N\N	N\N	N\N	N\N	N\N

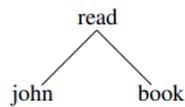
Both readings show the phrasal adnominal modifier iteration on_the_table under_the tree in_the_garden (TExer Sect. 5.1).

13. PARAPHRASE

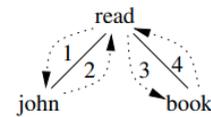
Speak mode paraphrase is the linguistic counterpart to hear mode ambiguity. Paraphrases of a content use the same semantic relations graph, but differ in their traversal order, as in the following example of the active-passive alternation (CLaTR Sect. 9.2):

13.1. ALTERNATION BETWEEN ACTIVE AND PASSIVE IN ENGLISH

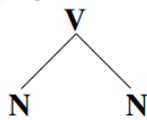
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature



(iv) surface realization

- a. 1 2 3 4
 John read a_book .
 V/N N/V V\N N\V
- b. 3 4 1 2
 A_book was_read by_John .
 V\N N\V V/N N/V

The active traverses the arcs of the NAG in the order 1, 2, 3, 4 (unmarked), while the passive traverses them in the order 3, 4, 1, 2 (marked).

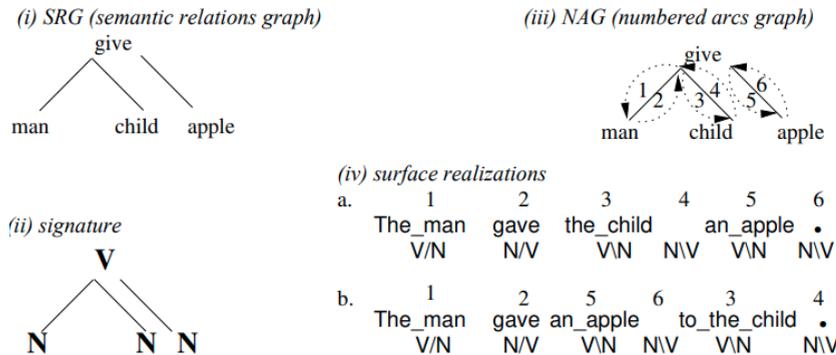
The content common to the speak mode derivation of the active and the passive requires a transitive verb and is defined explicitly as the following set of proplets:

13.2. CONTENT COMMON TO ACTIVE-PASSIVE VARIANTS

<p>[sur: john noun: [person x] cat: snp sem: nm m fnc: read mdr: nc: pc: prn: 2</p>	<p>[sur: verb: read cat: #n' #a' decl sem: ind past arg: [person x] book mdr: nc: pc: prn: 2</p>	<p>[sur: noun: book cat: snp sem: indef sg fnc: read mdr: nc: pc: prn: 2</p>
---	--	--

The difference between 'John read a book' (surface realization a) and 'A book was read by John' (surface realization b) is provided by the lexicalization rules embedded into the sur slot of the goal pattern of the navigation operations (17.6, 17.7). Another example of a speak mode paraphrase in English is an alternation involving the indirect and the direct object, as in 'The man gave the child an apple' vs. 'The man gave an apple to the child':

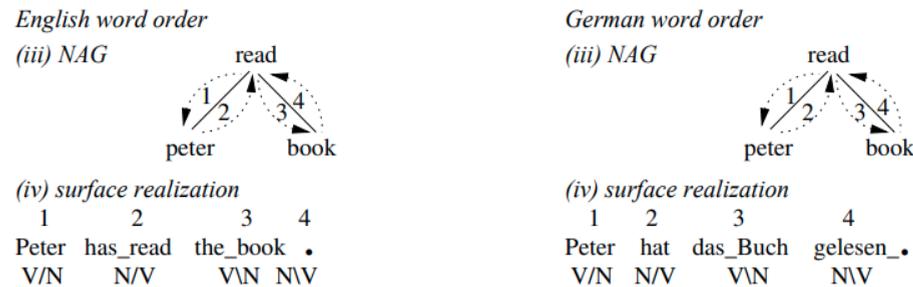
13.3. SURFACE ALTERNATION IN 3-PLACE VERB PROPOSITION



Variant a is based on the traversal order 1, 2, 3, 4, 5, 6, while variant b is based on the order 1, 2, 5, 6, 3, 4. The prepositional phrase 'to_the_child' is produced by the language-dependent lexicalization rule in the goal pattern of transition 3 of variant b.

While the examples of paraphrase in 13.1 and 13.3 are within the English language, there are similar kinds of paraphrase also between languages:

13.4. WORD ORDER DIFFERENCE BETWEEN ENGLISH AND GERMAN



The SRG, the signature, and the NAG are the same for English and German because they characterize semantic relations in the same content, independently of the language-dependent surface realization. The common content is defined as follows:

13.5. CONTENT OF 'PETER HAS READ THE BOOK.'

sur: peter noun: [person x] cat: snp sem: nm m fnc: read mdr: nc: pc: prn: 79	sur: verb: read cat: #n' #a' decl sem: perf arg: [person x] book mdr: nc: pc: prn: 79	sur: noun: book cat: snp sem: def sg fnc: read mdr: nc: pc: prn: 79
---	---	---

The word order, however, differs between the two languages. In English it is Peter has read the book. and in German Peter hat das Buch gelesen. (*Peter has the book read.*), as shown by the alternative surface realizations in 13.4.

In communication, the word order differences between two languages originate in the language-dependent lexicalization rules, which may segment surfaces differently in the traversal of arcs. For example, the complex verb form ‘has_read’ in English is realized in navigation step 2 and the period in step 4. In German, in contrast, ‘hat’ is realized in step 2 and ‘gelesen_’ in step 4 (Satzklammer).

14. COMPUTATIONAL COMPLEXITY AND GRAMMATICAL DISAMBIGUATION

According to Transformational Grammar, the computational complexity degree of natural language is *undecidable*⁴ (formal proof by Peters & Ritchie 1973). According to DBS, natural language is *linear* (formal proof in TCS’92). How is this possible?

The different computational complexity degrees do not apply to natural language *per se*, but to different theoretical reconstructions of natural language: one as some spurious “language acquisition device,” the other as a means of communication between cognitive agents speaking the same natural language.

The argument for the linear complexity degree of DBS is as follows: the grammars of the DBS hear, think, and think-speak mode are all C-Lags.⁵ In a C-Lag, the total number of elementary computational steps in any single operation application is below a grammar-dependent worst case limit (finite constant C). Thus, an individual DBS operation application alone cannot cause higher complexity. The only possibility to increase complexity is systematically repeating (recursive or iterative) ambiguity, i.e. the systematic generation of parallel readings.

In the speak mode, there are no parallel readings because there are no ambiguities, only paraphrases (Sect. 13). Alternative paraphrases result from the choice between continuations in the traversal of a given input content and is decided by the agent’s rhetorical purpose. Human speakers can not produce paraphrases simultaneously and human hearers could not process them simultaneously.

In the hear mode, a reading of length n requires exactly n-1 operation applications. The only way to increase complexity in DBS is a systematic multiplication of readings with the

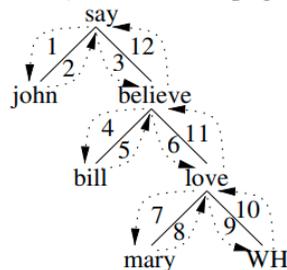
4 This has been noted to be unlikely by, among others, Harman (1963), Gazdar (1981), McCawley (1982), and Ross (1986). The claim of undecidability follows directly from the substitution-based (irregular) derivation order of TG, which fails to be input-output equivalent with the natural prototype.

5 TCS’92, p. 291, definition 4.1.

length of the hear mode input. Such a *recursive ambiguity* is illustrated in formal language analysis (FoCL 11.5.8, SubsetSum), but can not be found in natural language. The reason is *grammatical disambiguation*, as in the following example of what is known in linguistics as an ‘unbounded dependency’:

14.1. DBS GRAPH OF AN UNBOUNDED DEPENDENCY

(iii) NAG (numbered arcs graph)

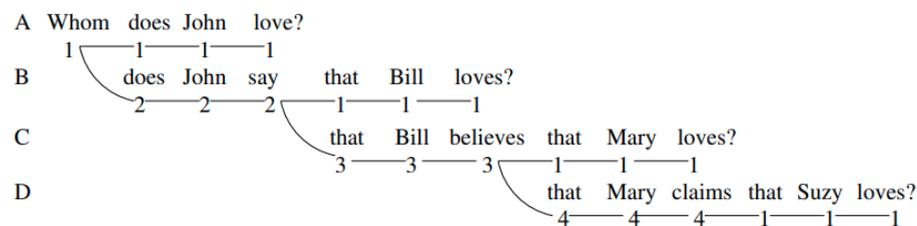


(iv) surface realization

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

There is no grammatical limit on the number of possible object sentences intervening between initial ‘Whom’ and the final clause, here ‘that Mary loves’. For example. ‘that Bill believes that Suzy said that Bob believes that Lucy said ...’ may theoretically be continued indefinitely with additional object clauses. However, with each new object clause there arises an obligatory choice between termination vs. continuation. For example, does John may be continued with love (no object clause, derivation terminates) or with say that (object clause, derivation continues). Because terminated ambiguities are discarded,⁶ i.e. do not add up in the result, the output is unambiguous:

14.2. AMBIGUITY STRUCTURE OF AN UNBOUNDED SUSPENSION



The time-linear cycle is terminated by the first verb which takes the initial Whom as its object. Because each continuation with another object clause $n+1$ removes the terminated

⁶ In the terminology of DBS these ambiguities are [-global] (FoCL Sect. 11.3).

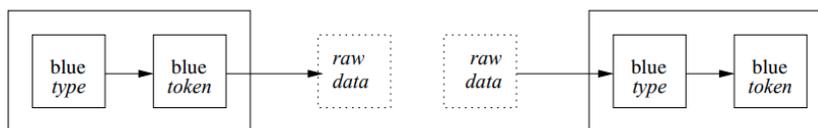
variant n from the set of readings (grammatical disambiguation), the construction is complexity-theoretically harmless. As long as there are no natural language examples of recursively or iteratively repeating [+global] ambiguities, the Linear Complexity Hypothesis for natural language stands without counterexample.

15. LANGUAGE COMMUNICATION

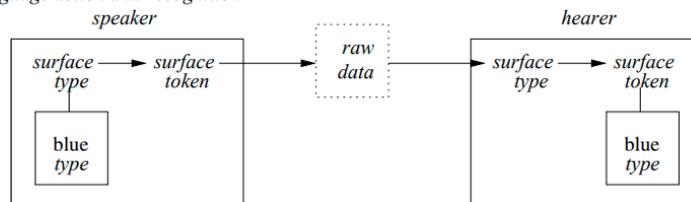
In primate evolution, nonlanguage cognition precedes language cognition. This raises the question of how nonlanguage cognition upscales to a natural language communication which is founded on cognition-external raw data. These data are without meaning or any grammatical properties whatsoever, but measurable by natural science:

15.1. COMBINING NONLANGUAGE INTO LANGUAGE COGNITION

(i) nonlanguage action and recognition



(ii) language action and recognition



The speak and the hear mode of (ii) language cognition reuse and combine the mechanisms of (i) nonlanguage recognition and action.

More specifically, nonlanguage and language cognition are alike in that they apply type-token matching to raw data input. They differ in that nonlanguage cognition applies type-token matching to nonlanguage content, while language cognition applies it to language surfaces. In the medium of speech, a surface token differs from its type by specifying volume, pitch, speed, timbre, etc., and in the medium of writing by specifying font, size, color, etc., i.e., what Aristotle would call the accidental (non-necessary) properties.

As an example of a language proplet with (i) a language-independent concept type and (ii) a language-dependent surface token consider the German word form *blaues*:

15.2. PROPLET WITH SURFACE TOKEN AND CONCEPT TYPE

sur: blaues	
adj:	place holder: blue sensory modality: vision semantic field: color content kind: concept wavelength: 490-450 nm frequency: 610-670 THz samples: a'', b'', c'', ...
cat: adn	
sem: pad n sg indef na	
mdd: car	
mdr:	
pc:	
nc:	
prn: 67	

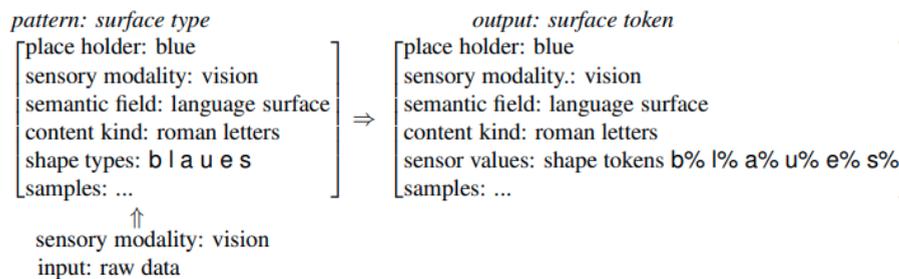
Type-token adaptation in speak mode surface production may be illustrated as follows:

15.3. SPEAK MODE; CONTENT TO SURFACE TYPE TO RAW DATA

<i>input: proplet token</i>	sur:	
	adj: blue	
	cat: adn	
	sem: pad	
	fnc: have	
	...	
	prn: 67	
	↓	
<i>output: surface type</i>	place holder: blue	⇒ %b %l %a %u %e %s raw output
	sensory modality: manipulation	
	semantic field: language surface	
	content kind: Roman letters	
	actuator values: type shapes b l a u e s	
	samples: ...	

The input, i.e. the content token *blue* of nonlanguage cognition, retrieves the corresponding language-dependent surface, here the type of German *b l a u e s*, based on a list which provides allomorphs using the input proplet's core, cat, and sem values. This output serves as input to a realization operation which adapts the surface type into a token, realized as raw data. Type-token instantiation in hear mode surface recognition may be illustrated as follows:

15.4. HEAR MODE: RAW DATA TO SURFACE TYPE TO CONTENT



The input consists of raw data which are provided by the agent's vision sensors and matched by the letters' shape types provided by the agent's memory. The output replaces the shape types, here *b l a u e s*, with the matching raw data resulting in shape tokens; shown as *b% l% a% u% e% s%*, they record the accidental properties. The function crucial for the understanding of the hearer, however, is using the place holder, here *blue*, for the lexical look-up of the correct nonlanguage concept in 7.5.

The purpose of producing surface tokens in the speak mode and of recognizing surface tokens in the hear mode is *content transfer*. For transfer to be successful, speaker and hearer must use the same surface-content pairs. This requires speaker and hearer to have learned (stored in memory) the (a) content, the (b) surface, and the (c) convention connecting (a) and (b) in language acquisition. Adhering to grammatical wellformedness is a functional precondition (Lewis 1969) for the successful transfer of content from speaker to hearer.

As a computationally effective reconstruction of natural language communication, 15.1 requires the agent-based data-driven ontology of DBS. The explanation of the evolutionary transition from nonlanguage to language cognition is in the spirit of Charles Darwin and out of reach for a sign-based substitution-driven ontology.

PART II: SOME TECHNICAL DETAILS

In DBS, the output of the speak mode and the input to the hear mode is an agentexternal sequence of word form surfaces in the form of raw data. Their strictly timeliner derivation order, imposed by input-output equivalence with the natural prototype, is called *left-associative* in computer science compiler construction.⁷ Left-associative composition adds one item after another at the end of a sequence and has the bracketing structure ((((((a b)c)d)e)f)... An unambiguous hear mode derivation consisting of *n* word form surfaces (length *n*) requires exactly *n*-1 left-associative compositions.

⁷ Aho and Ullman 1977, p. 47. Thanks to Profs. Ron Kaplan and Stuart Shieber for pointing it out.

Each operation application of the DBS think, think-speak, and hear mode is of constant complexity (C-LAGs). Therefore the only way to increase computational complexity in DBS is recursive or iterative ambiguity. Because this kind of ambiguity does not occur in natural language (Sect. 14, grammatical disambiguation), the computational complexity degree of natural language in DBS is *linear*.

16. THE FOUR OPERATION COMPONENTS OF DBS

As a model of natural language communication, DBS consists of four operation components, all of which use the same left-associative (time-linear) derivation order:

16.1. COMPONENTS USING LEFT-ASSOCIATIVE OPERATIONS

Hear mode syntax:

a sentence start combines with a next word into a new sentence start.

Hear mode morphology:

a word form start combines with a next allomorph into a new word form start.⁸

Speak mode navigation:

a start content combines with a next proplet into a new start content.

Speak mode surface realization:

a start surface combines with a next surface into a new start surface.

Left-associative operations are *incremental* and strictly *time-linear*.⁹

17. THE SEMANTIC RELATIONS OF STRUCTURE

In concord with the classical tradition in philology, DBS distinguishes four kinds of semantic relations of structure¹⁰ between proplets:

8 FoCL Sect. 14.4. Hermann Paul (1889, Chap. 19) calls morphology 'frozen syntax.' 11

9 The notion time-linear is a derivation order in linguistics, while linear is a complexity degree in computer science. In contrast to the substitution-based derivation order of PSG and TG (which is a partial order and results in polynomial and undecidable complexity, respectively), the time-linear derivation order of DBS is a total order (which results in linear complexity). The distinction between 'partial' vs. 'total' order comes from mathematics (number theory).

10 The linguistic counterpart to the semantic relations of structure are the semantic relations of the lexicon.

- (1) subject/predicate
- (2) object\predicate
- (3) modifier|modified
- (4) conjunct-conjunct.

The first three constitute the traditional notions of functor-argument, while the last is coordination. The first two are obligatory, while the latter two are optional.

DBS shows the four classical semantic relations of structure in **(a)** a graphical format and a **(b)** linear notation, and in the **(i)** hear and the **(ii)** speak mode.

17.1. (A, B) STATIC PRESENTATION OF THE 4 SEMANTIC RELATIONS

	subject/predicate	object\predicate	modifier modified	conjunct-conjunct
<i>graphical format</i>				
<i>linear notation</i>	N/V	N\V	A N	N-N

Extrapositional variants are V/V, V\N, V|V, V-V.

Convention: in the linear notation of functor-arguments, the lower node in the binary graph precedes.

The *direction* of traversing a semantic relation of structure (activating a content in the think or the think-speak mode) is indicated by numbered arcs in the graph and by extending the slashes into arrows in linear notation, e.g. V\N and N\V:

17.2. (i, ii) DYNAMIC PRESENTATION OF THE 4 SEMAN. RELATIONS

	subject/predicate	object\predicate	modifier modified	conjunct-conjunct
<i>graphical format</i>				
<i>linear notation:</i>	V↓N	V↓N	N↓A	forward: N→N
<i>notation:</i>	N↑V	N↑V	A↑N	backward: N←N

Convention: in the linear notation of functor-arguments, the start node precedes the goal node.

The following examples compare (i) a content in the DBS proplet format (set of proplets concatenated by semantic relations of structure, coded by address) with (ii) the equivalent DBS graph analysis:¹¹

17.3. FORMAT I: CONTENT OF JOHN WASHED THE NEW CAR.

[sur: john noun: [person x] cat: snp sem: nm m fnc: wash mdr: nc: pc: prm: 1]	[sur: verb: wash cat: #n' #a' decl sem: ind past arg: [person x] car mdr: nc: pc: prm: 1]	[sur: noun: car car: snp sem: def sg fnc: wash mdr: new nc: pc: prm: 1]	[sur: new adj: new cat: adn sem: pad mdd: car mdr: nc: pc: prm: 1]
--	--	--	---

The corresponding DBS graph analysis presents the content in four views, which use the graphical connectives /, \, |, and -, and may be listed as follows:

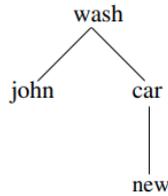
17.4. THE FOUR VIEWS OF A DBS GRAPH ANALYSIS

- (i) the semantic relations graph (SRG)
 - uses the core values of a content
- (ii) the signature
 - uses the core attributes of a content
- (iii) the numbered arcs graph (NAG)
 - adds numbered arcs to the SRG.
- (iv) the surface realization
 - shows three parallel lines with each segment indicating
 - (a) the number of the arc traversed
 - (b) the language-dependent surface produced
 - (c) the operation name of the transition.

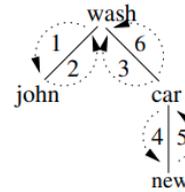
¹¹ A third graphical format is the time-linear hear mode derivations, which are rather bulky but highly informative. See TExer 2.1.2, 2.1.6, 2.2.2, etc., for more than 24 grammatical constructions of English.

17.5. FORMAT II: SEMANTIC RELATIONS GRAPH FOR 17.3

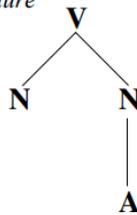
(i) SRG (semantic relations graph)



(iii) NAG (numbered arcs graph)



(ii) signature

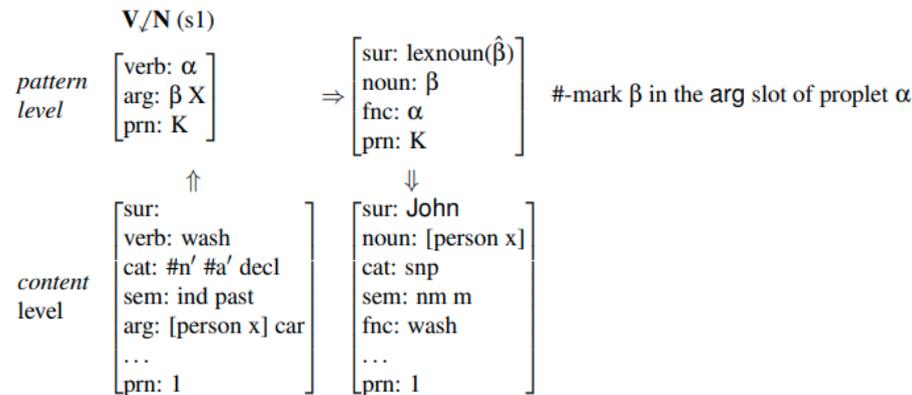


(iv) surface realization

1	2	3	4	5	6
John	washed	the	new	car	.
V/N	N/V	V/N	N/A	A/N	NV

The numbered arcs of the NAG are traversed by think-speak operations which take a single start proplet as input and produce a single goal proplet as output:

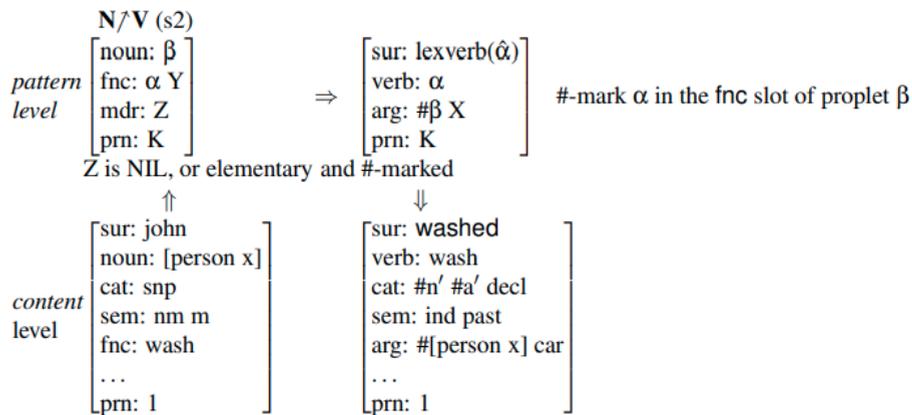
17.6. DOWNWARD TRAVERSAL WITH A THINK-SPEAK OPERATION



This operation application shows the downward traversal from the predicate to the subject via arc 1 in the NAG of 17.5. The operation name, here V/N, is followed by the operation number, here (s1), referring to the DBS speak mode grammar in TExer 6.5.1. In DBS, the content kind of *name* provides automatic word form production with the feature [sur: lexnoun($\hat{\beta}$)], which uses and overwrites the name marker in the goal proplet (CASM).

The corresponding upward traversal (arc 2 in 17.5) is provided by N/V:

17.7. UPWARD TRAVERSAL WITH A THINK-SPEAK OPERATION



Automatic word form production is based on the function $\text{lexverb}(\hat{\alpha})$, which uses the core value wash and the sem values ind past to realize wash-ed.

18. HEAR MODE: AUTOMATIC WORD FORM RECOGNITION

The raw data of the language surfaces transferring content from speaker to hearer occur in the following media (CC Sect. 11.2):

18.1. MEDIA OF NATURAL LANGUAGE COMMUNICATION

- (i) spoken,
- (ii) written (including Braille), and
- (iii) signed

The speak mode requires the input of content to produce language-dependent surfaces. The hear mode requires the input of language-dependent surfaces to produce content. The crucial problem of automatic word form recognition in the medium of speech is the segmentation of the input stream into word forms, e.g. continua ‘theolddoglooked...’, into ‘the+old+dog+looked+...’, and the word forms into allomorphs, e.g. ‘looked’ into ‘look-ed’, for such semantic distinctions as number, tense, and mood.

In the medium of speech, voice independent segmentation and allomorph lookup are the tasks of automatic speech recognition (ASR), which is largely based on statistics. In the medium of writing, used by DBS, the solution is based on a trie structure, string search, and a lexicon of allomorphs (FoCL Chap. 14).

Automatic word form recognition and production is the first obligatory step for any computational linguistic analysis of a natural language. Even when restricted to written language (which facilitates the task), it can keep a medium-sized team of researchers busy for decades, and then requires continuous maintenance. It involves not only recognition and production of surfaces, but also their lexical analysis. The programming is a major investment, but of solid long-term theoretical and practical use by many parties.

19. HEAR MODE: MAPPING SURFACES INTO CONTENT

For development and maintenance, DBS word form lookup may be run in isolation. The input are huge lists of word forms, automatically produced from categorized base forms and the associated paradigms. The output is a list of unconnected word form proplets:

19.1. ISOLATED LOOKUP OF LEXICAL PROPLETS

[sur: John noun: [person x] cat: snp sem: nm m fnc: mdr: nc: pc: prn:	[sur: wash-ed verb: wash cat: n' a' v sem: ind past arg: mdr: nc: pc: prn:	[sur: the noun: n_1 cat: sn' snp sem: def sg fnc: mdr: nc: pc: prn	[sur: new adj: new cat: adn sem: pad mdd: mdr: nc: pc: prn	[sur: car noun: car cat: sn sem: sg fnc: mdr: nc: pc: prn:	[sur: • verb: v_1 cat: v' decl sem: arg: mdr: nc: pc: prn:
---	--	--	--	--	--

The method of choice is the allomorph approach (FoCL Chap. 14).

In contradistinction to isolated lookup, word form lookup in the hear mode is incrementally intertwined with syntactic-semantic composition:

19.2. LEFT-ASSOCIATIVE (INCREMENTAL) LEXICAL LOOKUP

Sentence Start + Next Word ⇒ Next Sentence Start

John washed+the ⇒ John washed the

John washed the+new ⇒ John washed the new

John washed the new+car ⇒ John washed the new car etc.

Instead of the generic connective '+', syntax-semantics in the DBS hear mode uses three differentiated connectives:

- × for cross-copying
- U for absorption
- ~ for suspension

The hear mode derivation taking the sequence of lexical proplets in 19.1 as input results in the following content:

19.3. CONTENT OF JOHN WASHED THE NEW CAR.

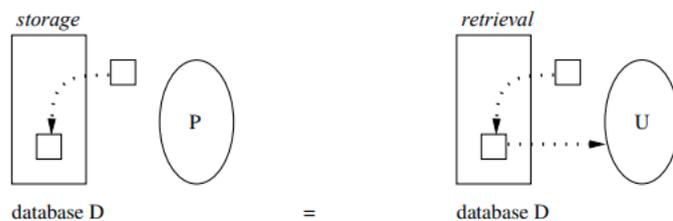
sur: john noun: [person x] cat: snp sem: nm m fnc: wash mdr: nc: pc: prn: 1	sur: verb: wash cat: #n' #a' decl sem: ind past arg: [person x] car mdr: nc: pc: prn: 1	sur: noun: car car: snp sem: def sg fnc: wash mdr: new nc: pc: prn: 1	sur: new adj: new cat: adn sem: pad mdd: car mdr: nc: pc: prn: 1
---	---	---	--

This content differs from the lexical lookup 19.1 in that the proplets are connected by semantic relations of structure, coded by address.

20. STORAGE AND RETRIEVAL OF CONTENT IN DBS DATABASE

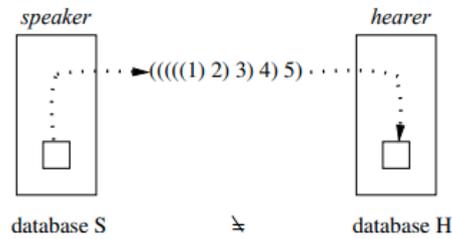
An essential component of DBS is the agent's on-board memory, defined as a contentaddressable database. Consider the schematic comparison of 20.1 and 20.2:

20.1. CONVENTIONAL DATABASE INTERACTION



Interaction takes place between different agents using the same external database and the same artificial language, e.g. SQL: programmer P controls the storage and the user U controls the retrieval operations.

20.2. SPEAKER AND HEARER INTERACTING IN COMMUNICATION



Interaction takes place between different agents using different on-board databases and the same natural language. The transfer of content from speaker to hearer is completely automatic and based on agent-external raw data. In the speak mode, automatic word form production takes cognitive content as input and maps it into language-dependent surfaces as raw data output. In the hear mode, automatic word form recognition takes raw data as input and maps it into cognitive content as output (turn-taking). The content 19.3 is stored alike in both agents' DBS databases as follows:

20.3. DATABASE SCHEMA OF CONTENT-ADDRESSABLE A-MEMORY

<i>member proplets</i>	<i>now front</i>	<i>owner values</i>
<pre> [sur: noun: car car: snp sem: def sg ... fnc: wash mdr: new nc: pc: prn: 1] </pre>		car
...		...
<pre> [sur: john noun: [person x] cat: snp sem: nm m ... fnc: wash mdr: nc: pc: prn: 1] </pre>		john =[person x]
...		...

	[sur:	
	adj: new	new
	cat: adn	
	sem: pad	
...	mdr: car	
	nc:	
	pc:	
	prn: 1	
	...]	...
	[sur:	
	verb: wash	wash
	cat: #n' a' v	
	sem: ind past	
...	arg: [person x] car	
	mdr:	
	nc:	
	pc:	
	prn: 1	
	...]	

Horizontally, the DBS database schema is token lines as lists of proplets with the same core value, stored in the temporal order of arrival. Vertically, the token lines are in the alphabetical order induced by their proplets' core value. The schema is contentaddressable because it does not use a separate index (unlike an RDBMS).

The column of owners provides access to the token line of proplets to be stored or retrieved. In recognition, proplets are stored at the now front in the token line of their core value. In action, content is activated by navigating along the semantic relations between proplets, using the address for retrieval of the goal proplet. Because the semantic relations between proplets are coded by address, proplets are order-free: the semantic relations of structure between them hold regardless of where they are located.

Within the token lines, the field of *member proplets* is the permanent memory which may never be changed. The only way to correct is by adding new content, as in a diary. The *now front* is the arena of current content processing. It is cleared in regular intervals by moving it (together with the owners) into fresh memory territory, leaving its content behind in what is becoming permanent member proplets (loom-like clearance).

PART III: DATA COVERAGE

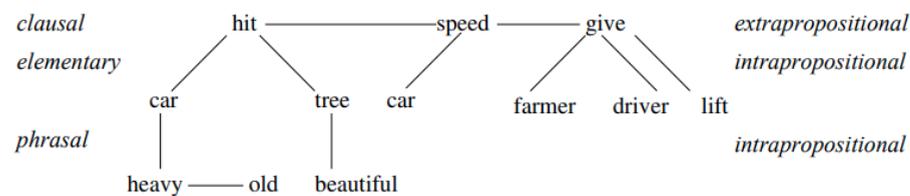
The most detailed and extensive DBS data coverage for English so far is provided in TExer (318 pp.). Here we conclude with a brief overview at a high level of abstraction.

The expressions of natural language occur at three levels of grammatical complexity, traditionally called (i) elementary, (ii) phrasal, and (iii) clausal. Phrasal contents are built from

elementary contents, and clausal contents from elementary and phrasal contents. A text is analyzed as a conjunction of sentential clausal contents.

The following SRG shows an extrasentential coordination. It is a three sentence text, analyzed as a conjunction of three propositions, each with elementary arguments, and one with two phrasal arguments:

20.4. CLAUSAL, ELEMENTARY, AND PHRASAL RELATIONS

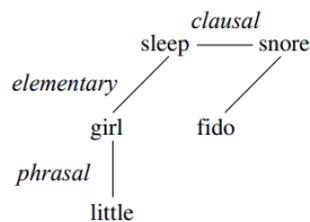


For explicit derivations see NLC, Chap. 13 (hear mode) and Chap. 14 (speak mode).

The next example compares two intrasentential constructions consisting of the same component contents, but connected in one by extrapositional coordination (parataxis) and the other by subordination (hypotaxis):

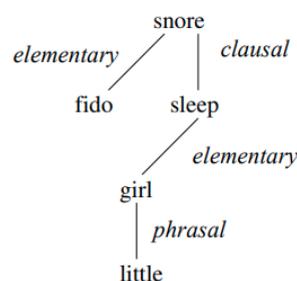
20.5. CLAUSAL COORDINATION VS. MODIFICATION

horizontal connection (conjunct–conjunct)



Corresponding English surfaces:
The little girl slept. Fido snored.

vertical connection (modifier|modified)

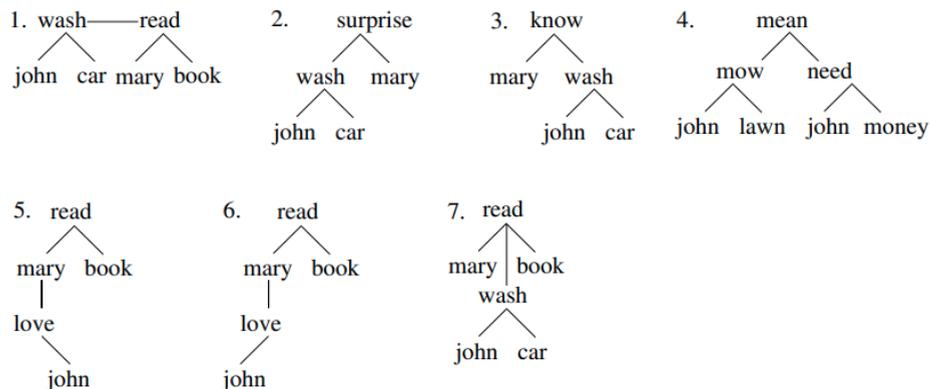


When the little girl slept, Fido snored.

As contents, the constructions are semantically similar, but syntactically different. The variety of extrapositional constructions results from the four different kinds of semantic relations of structure in natural language and is reflected graphically by the semantically interpreted /, \, |, and - lines (edges).

Finally consider the following variety of extrapositional structures:

20.6. RELATING TWO TRANSITIVE VERBS EXTRAPROPOSITIONALLY



1. conjunct—conjunct: John washed the car. Mary read a book.
2. subject/predicate: That John washed the car surprised Mary.
3. object\predicate: Mary knew that John had washed the car.
4. sbj/prd\obj: That John mows the lawn means that John needs money.
5. adn_mdr|mdd (subject gap): Mary who loves John read a book.
6. adn_mdr|mdd (object gap): Mary who(m) John loves read a book.
7. adv_mdr|mdd: When John washed the car Mary read a book.

Which kind of relation may connect the predicates of two component propositions depends on the verb class (Levin 2009).

21. CONCLUSION

In contradistinction to PSG, the DBS analysis of natural and formal languages does not use any non-terminal nodes. Instead, the nodes in a grammatical structure analysis are proplets, defined as nonrecursive feature structures with ordered attributes which take values of grammar and of content.

Instead of connecting nonterminal nodes with the notions of dominance and precedence (which are more appropriate for the social domain of pomp and circumstance than natural language semantics), DBS uses the classical semantic relations of subject/predicate, object\predicate, modifier|modified, and conjunct—conjunct.

In contradistinction to substitution-driven systems like PSG and CG, DBS is inputoutput equivalent with the natural prototype. As a result, the computational complexity of DBS is of linear degree, in contrast to context-free Phrase Structure Grammar and Categorical Grammar, which are polynomial, and to Transformational Grammar, which is undecidable.

REFERENCES

- Aho, A.V. and J.D. Ullman (1977). *Principles of Compiler Design*, Reading, Mass.: Addison-Wesley.
- Aristotle, *Analytica Priora*, in Aristoteles Latinus III.1–4, L. Minio-Paluello (ed.), Bruges–Paris: Desclée de Brouwer, 1962.
- CASM = Hausser, R. (2017) “A computational treatment of generalized reference,” *Complex Adaptive Systems Modeling*, Vol. 5.1:1–26. Also at lagrammar.net and <http://link.springer.com/article/10.1186/s40294-016-0042-7>.
- CC = Hausser, R. (2019) *Computational Cognition: Integrated DBS Software Design for Data-Driven Cognitive Processing*, pp. i–xii, 1–237, lagrammar.net .
- CLaTR = Hausser R. (2011) *Computational Linguistics and Talking Robots: Processing Content in DBS*, pp. 286. Springer (preprint 2nd ed. at lagrammar.net).
- FoCL = Hausser, R. (1999) *Foundations of Computational Linguistics, Human–Computer Communication in Natural Language, 3rd ed. 2013*, Springer.
- Frege, G. (1892) “Über Sinn und Bedeutung,” *Zeitschrift für Philosophie und philosophische Kritik*, Vol. 100:25-50.
- Gazdar, G. (1981) “Unbounded Dependencies and Coordinate Structure,” *Linguistic Inquiry* Vol. 12.2:155–184.
- Harman, G. (1963) “Generative Grammar without Transformational Rules: a Defense of Phrase Structure,” *Language* Vol. 39:597–616.
- JHaspelmath, M., and S.M. Michaelis (2017) “Analytic and synthetic: Typological change in varieties of European languages,” In: Isabelle Buchstaller & Beat Siebenhaar (eds.), *Language variation – European perspectives VI: Selected papers from the 8th International Conference on Language Variation in Europe (ICLaVE 8)*, Leipzig 2015. Amsterdam: Benjamins.
- Hubel, D.H., and T.N. Wiesel (1962) “Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat’s Visual Cortex,” *Journal of Physiology*, Vol. 160:106–154.
- Kripke, S. (1972) “Naming and Necessity,” in D. Davidson and G. Harman (eds.), 253–355 Levin, B. (2009) “Where Do Verb Classes Come From?” <http://web.stanford.edu/~bclevin/ghent09vclass.pdf>.
- Lewis, D. (1969). *Convention: a Philosophical Study*, Hoboken, N.J.: Wiley-Blackwell Liebermann, P. (1984) *The Biology and Evolution of Language*, Cambridge Mass.: Harvard University Press.
- Liebermann, P. (2000) *Human Language and our Reptilian Brain, The Subcortical Basis of Speech, Syntax, and Thought*, Cambridge Mass.: Harvard University Press.
- Neisser, U. (1967) *Cognitive Psychology*, New York: Appleton-Century-Crofts.
- NLC = Hausser R. (2006) *A Computational Model of Natural Language Communication: Interpretation, Inference, and Production in DBS*, pp. 360. Springer (preprint 2nd ed. at lagrammar.net).
- Neumann, G. (1998) “Ambiguity vs. Paraphrasing” <https://www.dfki.de/~neumann/publications/diss/node47.html>
- Paul, H. (1889) *Prinzipien der Sprachgeschichte*, 2nd Ed., Halle a.S.: Max Niemeyer.
- Peirce, C.S. (1903) “Lectures on Pragmatism”, CP 5.171.
- Peirce, C.S. (1931–1935) CP (*Collected Papers*), C. Hartshorne and P. Weiss (eds.), Cambridge, MA: Harvard Univ. Press.
- Peters, S., and R. Ritchie (1973) “On the Generative Power of Transformational Grammar,” *Information and Control* Vol. 18:483–501.
- Pylshyn, Z. (2009) “Perception, Representation and the World: The FINST that binds,” in D. Dedrick & I.M. Trick (eds.) *Computation, Cognition and Pylshyn*, Cambridge MA: MIT press.

Ross, J. R. (1986) *Infinite syntax!*, Norwood, NJ: ALEX, Russell, B. (1905) "On denoting," *Mind*, Vol. 14:479-493.

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

Sperling, G. (1960) "The Information available in Brief Visual Processing," *Psychological Monographs*, 11, Whole No. 498.

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

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

Wiriyathamabhum, P., D. Summers-Stay, C. Fermüller, and Y. Aloimonos (2016) "Computer Vision and Natural Language Processing: Recent Approaches in Multimedia and Robotics," December 2016, *ACM Computing Surveys* 49(4):1-44