

Comparing DBS and Predicate Calculus

Roland Hausser

Universität Erlangen-Nürnberg (em.)

©Roland Hausser, January 6, 2021

Abstract

A content like *Fido found a bone.* may be viewed from two basic perspectives. One takes the view of an outside observer. For example, Predicate Calculus models the truth relation between a formal sign and a formal model (Sect. 1) or model structure (Sect. 8). This approach is called *sign-based*.

The other takes the view of an agent, i.e. Fido looking, listening, and sniffing out into the world, keeping track of it, and deriving suitable action. For example, DBS models the cognition of an artificial agent with on-board interface and memory components for automatically monitoring recognition (Sect. 11) and action (Sect. 12), and for storing and retrieving content. Because the agent's processing of sensory input and output is central to this approach, it is called *agent-based*.¹

1 Definition of Predicate Calculus

Montague's (1973) version of first order Predicate Calculus, hence PredC, is widely admired. Leaving the intension-extension distinction and the use of lambda calculus aside, his formalization may be presented as follows (slightly revised):

1.1 FORMAL DEFINITION OF PREDC

1. A model \mathcal{M} is defined as the ordered quadruple $\langle A, B, F, g \rangle$, where A is an infinite set of objects or individuals, B a finite set of basic expressions, F a denotation function from B into the free monoid² A^* over A , and g an assignment function from variables into A^* .
2. The elements \emptyset (empty set) and $\{\emptyset\}$ (set containing the empty set) of A^* are used as denotations of the truth values 1 (true) and 0 (false), respectively.
3. Syntactically, the set of operators $\neg, \wedge, \vee, =, \rightarrow, \forall, \exists$ is defined as follows:
 - (a) If f is a one-place functor and α is a name, then $f(\alpha)$ is a sentence.
 - (b) If ϕ is a sentence, then $\neg\phi$ is a sentence.
 - (c) If ϕ is a sentence and ψ is a sentence, then $\phi \wedge \psi$ is a sentence.
 - (d) If ϕ is a sentence and ψ is a sentence, then $\phi \vee \psi$ is a sentence.
 - (e) If ϕ is a sentence and ψ is a sentence, then $\phi \rightarrow \psi$ is a sentence.
 - (f) If ϕ is a sentence and ψ is a sentence, then $\phi = \psi$ is a sentence.
 - (g) If f and h are functors and x is a variable, then $\exists x[f(x) \wedge h(x)]$ is a sentence.
 - (h) If f and h are functors and x is a variable, then $\forall x[f(x) \rightarrow h(x)]$ is a sentence.

Definitions (a–f) constitute Propositional Calculus, hence PropC. Definitions (g–h) extend PropC into PredC by introducing the quantifiers $\exists x$ and $\forall x$ binding a variable. Within a formula, the *scope* of a quantifier is the area in which its variable is bound. The area is defined by the formula's bracketing structure.

¹For attempts at integrating symbolic logic into an agent-based cognition see Hausser (1980), Kamp (1980), and Barwise and Perry (1983).

² A^* serves as the "universe of discourse." The $*$ is called the Kleene Star.

4. Semantically, the set of operators is defined as follows:

- (a) If f is a one-place functor, then $f(\alpha)$ is a true sentence relative to a model \mathcal{M} if and only if the denotation of α in \mathcal{M} is an element of the denotation of f in \mathcal{M} .
- (b) $\neg \phi$ is a true sentence relative to a model \mathcal{M} if and only if the denotation of ϕ is 0 relative to \mathcal{M} .
- (c) $\phi \wedge \psi$ is a true sentence relative to a model \mathcal{M} if and only if the denotations of ϕ and of ψ are 1 relative to \mathcal{M} .
- (d) $\phi \vee \psi$ is a true sentence relative to a model \mathcal{M} if and only if the denotation of ϕ or ψ is 1 relative to \mathcal{M} .
- (e) $\phi \rightarrow \psi$ is a true sentence relative to a model \mathcal{M} if and only if the denotation of ϕ is 0 relative to \mathcal{M} or the denotation of ψ is 1 relative to \mathcal{M} .
- (f) $\phi = \psi$ is a true sentence relative to a model \mathcal{M} if and only if the denotation of ϕ relative to \mathcal{M} equals the denotation of ψ relative to \mathcal{M} .
- (g) $\exists x[f(x) \wedge h(x)]$ is a true sentence relative to \mathcal{M} and a variable assignment \mathbf{g} if and only if at least one $\mathbf{g}'(x)$ makes $[f(x) \wedge h(x)]^{\mathcal{M}, \mathbf{g}}$ true.
- (h) $\forall x[f(x) \rightarrow h(x)]$ is a true sentence relative to \mathcal{M} and a variable assignment \mathbf{g} if and only if all $\mathbf{g}(x)$ make $[f(x) \rightarrow h(x)]^{\mathcal{M}, \mathbf{g}}$ true.

Today's PredC originated with Frege (1879). Montague complemented PredC by using lambda-reduction, the intension-extension dichotomy, and the assignment function \mathbf{g} .

2 PredC Overgeneration

The syntactic rules of PropC and their semantic interpretation rely on recursion. For example, if $(p \wedge q)$ and r are sentences, then $((p \wedge q) \wedge r)$, $((p \wedge q) \vee r)$, $((p \wedge q) \rightarrow r)$, and $((p \wedge q) = r)$ as well as $(r \wedge (p \wedge q))$, $(r \vee (p \wedge q))$, $(r \rightarrow (p \wedge q))$, and $(r = (p \wedge q))$ are sentences as well. The (i) order of the sub-sentences does not make a difference semantically (symmetry) except for the operator ' \rightarrow ' (asymmetry), the (ii) sentences resulting from the variations all have well-defined semantic interpretations relative to \mathcal{M} , and (iii) there is no upper bound on the number of variations because there is no limit on the length of PredC formulas.

Similarly in $\forall x[f(x) \rightarrow h(x)]$ and $\exists y[i(y) \wedge j(y)]$ of PredC, except that (i) the interpretation relative to \mathcal{M} requires the assignment \mathbf{g} , (ii) the variables p, q, \dots may be replaced by functors taking one or more quantified variables as argument, e.g. $f(x, y)$ or $f(x, y, z)$, and (iii) alternative substitution orders may produce systematic ambiguity:

2.1 ALLEGED AMBIGUITY OF Every dog finds a bone

1. $\forall x[\text{dog}(x) \rightarrow \exists y[\text{bone}(y) \wedge \text{find}(x, y)]]$
Every dog finds a bone
2. $\exists y[\text{bone}(y) \wedge \forall x[\text{dog}(x) \rightarrow \text{find}(x, y)]]$
There exists a bone which every dog finds

Linguistically, i.e. for the purpose of modeling natural language meaning with PredC (as in PSG), reading (1) is a correct representation of the meaning but reading (2) is not. Without being a genuine paraphrase, the PrepC structure of reading (2) is an artifact of the formalism. Because the alleged ambiguity occurs whenever the \exists and \forall quantifiers appear in the same formula, it is a systematic overgeneration of PredC.

3 Determiners in DBS

The DBS alternative to the PredC formulas in 2.1 is a set of proplets connected by semantic relations coded by address. Proplets are the computational data structure of DBS and defined as nonrecursive feature structures with ordered attributes.

3.1 UNAMBIGUOUS DBS ANALYSIS OF Every dog finds a bone

[sur: noun: dog cat: snp sem: pl exh fnc: find ... prn: 4	[sur: verb: find cat: #ns3 #a decl sem: pres arg: dog bone ... prn: 4	[sur: noun: bone cat: snp sem: indef sg fnc: find ... prn: 4
---	---	--

As in natural language, but unlike 2.1, there is no ambiguity, no coordination, and no coreference. The determiner aspect of $\forall x$ representing **every** in 2.1 is coded as the features [sem: exh pl] (exhaustive plural) and [cat: snp] (singular noun phrase) of the *dog* proplet, while that of $\exists y$ representing **a(n)** is coded as the feature [sem: indef sg] (indefinite singular) of the *bone* proplet. The semantic relations between the proplets *dog* and *find* are subject/predicate, and between *find* and *bone* object/predicate.

4 PredC Undergeneration

From a linguistic point of view, the counterpart to PredC overgeneration is PredC undergeneration, i.e. certain meaningful grammatical constructions which cannot be properly expressed. The classic example is the *donkey sentence*³: **Every farmer who owns a donkey beats it** (Geach 1962). A PredC derivation based on systematic substitution results in the following representation:

4.1 INCORRECT ANALYSIS OF WELL-FORMED SENTENCE IN PREDC

Every farmer who has a donkey beats it
 $\forall x[[\text{farmer}(x) \wedge \exists y[\text{donkey}(y) \wedge \text{own}(x,y)]] \rightarrow \text{beat}(x,y)]$

The English sentence is well-formed and meaningful, and the formal derivation is correct, but the formula is inadequate because the variable y in $\text{beat}(x,y)$ is not in the scope of the quantifier $\exists y$ binding $\text{donkey}(y)$.⁴

5 Coreference by Address

Alternative to treating coreference by means of quantifiers binding variables, DBS treats all semantic relations of structure, including coreference (CC Sect. 6.5), by means of proplet-internal address values.

³The quantifier scope problem manifested by the donkey sentence was recognized in the middle ages (Walter Burley 1328/Gualterus Burlaeus 1988). It is one of several instances in which PredC can not provide the linguistically correct quantifier scope. Discourse Representation Theory (DRT, Kamp 1980, Kamp and Reyle 1993) attempted to solve the problem of the donkey sentence while trying to maintain the sign-based substitution-driven foundation of PredC.

⁴There have been numerous proposals to avoid the “dangling variable” by fronting the existential quantifier. In this way, the y in $\text{donkey}(y)$ would get bound by $\exists y$ but at the cost of losing compositionality (King and Lewis 2017), which is methodologically unacceptable.

5.1 INTERPRETATION OF THE DONKEY SENTENCE IN DBS

noun: farmer	verb: own	noun: donkey	verb: beat	noun: (donkey 17)
cat: snp	cat: #n' #a' v	cat: snp	cat: #ns3' #a' decl	cat: snp
sem: pl exh	sem: pres	sem: indef sg	sem: pres	sem: sg
fnc: beat	arg: 0 donkey	fnc: own	arg: farmer (donkey 17)	fnc: beat
mdr: (own 17)	mdd: (farmer 16)	mdr:	mdr:	mdr:
nc:	nc:	nc:	nc:	nc:
pc:	pc:	pc:	pc:	pc:
[prn: 16]prn: 17]prn: 17]prn: 16]prn: 16

The grammatical embedding is reflected by the prn values, which are automatically incremented from 16 to 17 in the transition from the main to the sub-clause and decremented from 17 to 16 in the transition back to the main clause. The coreferential pronoun it in the main clause is represented by the extrapositional address (donkey 17), which refers to the antecedent⁵ in the sub-clause.

6 Propositions Denote Truth Values in PredC

An important difference between DBS and PredC is the treatment of concepts like *dog*, *small*, *find*, *bone*, and *big*. In PredC, concepts are treated as elementary mini-propositions which denote truth values relative to a set-theoretic model. Formally defined as functors which may differ in the number of arguments, e.g. $f(x)$ vs. $f(x,y)$, they are connected by the sentential operators of PropC and the quantifiers of PredC:

6.1 NOUN, VERB, AND ADJ FLATTENED INTO MINI-PROPOSITIONS

- the noun *dog* is interpreted as x is a *dog* and written as $\text{dog}'(x)$
- the adj *little* is interpreted as x is *little* and written as $\text{little}'(x)$
- the 1-place verb *snore* is interpreted as x *snore*s and written as $\text{snore}'(x)$
- the 2-place verb *find* is interpreted as x *find*s y and written as $\text{find}'(x, y)$
- the 3-pl. verb *give* is interpreted as x *gives* y z and written as $\text{give}'(x, y, z)$

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

6.2 PREDC REPRESENTATION OF *The little dog found a big bone*

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

The meaning difference between the constants *dog*, *little*, *bone*, *big*, and *find* depends on the denotation function F and the assignment function g , provided they are defined by the logician, which is usually not the case. The reason may be shown by explicitly defining a possible model for a PredC formula:

6.3 MINIMAL MODEL FOR THE PREDC FORMULA 6.2

Let \mathcal{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: $A = \{a_1, a_2, a_3\}$

⁵In linguistics, the term 'antecedent' is used not only for inferences, but also for the full noun referent, here a *donkey*, preceding a coreferential pronoun, here it (CLaTR Chap. 11).

$B = \{\text{dog, small, big, bone, eat}\}$
 $F(\text{dog}) = a_1,$
 $F(\text{small}) = a_1,$
 $F(\text{big}) = a_2,$
 $F(\text{bone}) = a_2,$
 $F(\text{eat}) = \langle a_1 a_2 \rangle,$
 $g(x) = a_1,$
 $g(y) = a_2$

Based on definitions (3) and (4) in 1.1, the formula 6.2 is well-formed and true in \mathcal{M} . However, if we defined $F(\text{dog})$ as a_3 , for example, the formula would be false.

In summary, because PredC’s pre-computational ontology provides neither a memory nor an interface component for autonomous recognition and action, formal models like 6.3 must be defined by hand. Also, a complete modeling of even the tiniest part of the world is out of reach. Therefore, explicit models are practically never defined. Instead logicians circumscribe models by means of conditionals which encode their intuitions about truth. DBS, in contrast, treats the world surrounding the cognitive agent *as given*⁶ and limits itself to automatically monitor the agent’s recognition and action computationally, using the agent’s on-board interface and memory components.

7 Propositions are Content in DBS

From a linguistic of view, treating nouns, verbs, and adjs uniformly as minipropositions (6.1) is semantically misguided because (i) it loses the classical distinction between referents, properties, and relations (CC 1.5.3), and (ii) obscures the empirical fact that relations and properties do not refer. Also, the use of coordination and coreference in connection with the quantifiers \forall and \exists (iii) violates the methodological standard of surface compositionality (FoCL Sect. 4.5) because there is neither coordination nor coreference in natural language expressions such as *The little dog found a big bone*.

DBS, in contrast, differentiates concepts into the three semantic kinds *referent*, *property*, and *relation*, with the syntactic correlates of elementary *noun*, *adj*, and *verb* (CC Sect. 1.5). Also, instead of propositions “denoting truth-values,” as in PropC and PredC, propositions ‘are content’ in DBS:

7.1 DBS INTERPRETATION OF *The little dog found a big bone*

$\left[\begin{array}{l} \text{sur:} \\ \text{noun: dog} \\ \text{cat: def sg} \\ \text{sem: animal} \\ \text{fnc: find} \\ \text{mdr: little} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 47} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{adj: little} \\ \text{cat: adn} \\ \text{sem: pad} \\ \text{mdd: dog} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 47} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{verb: find} \\ \text{cat: \#n' \#a' decl} \\ \text{sem: ind past} \\ \text{arg: dog bone} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 47} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{adj: big} \\ \text{cat: adn} \\ \text{sem: pad} \\ \text{mdd: bone} \\ \text{mdr:} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 47} \end{array} \right]$	$\left[\begin{array}{l} \text{sur:} \\ \text{noun: bone} \\ \text{cat: def sg} \\ \text{sem: body part} \\ \text{fnc: find} \\ \text{mdr: big} \\ \text{nc:} \\ \text{pc:} \\ \text{prn: 47} \end{array} \right]$
---	---	---	---	--

The proplets of a content are a set (order-free), held together by (i) a common prn value, here 47, and (ii) the semantic relations of structure coded by address, e.g. extrapositional (table 47) or intrapositional table (using the prn number of the proposition).

⁶Brooks (1986): “The world is its own best model.”

In addition to coding a semantic relation from one proplet to another, an address specifies a proplet's unique *storage location* (CC 12.4.4) in the agent's memory (CC Sect. 2.3). The core value of the address is used for finding the token line (vertical) and the *prn* value for finding the position in the token line (horizontal). Computationally, the token line is found by the letter sequence using string search (Knuth et al. 1977) in combination with a trie structure; the position in a token line is found using hashing.

8 Extending PredC to Possible Worlds

To accommodate the classical modalities of necessity (\Box) and possibility (\Diamond) as well as change in time and space, Montague extended models like 6.3 into model structures by adding the infinite sets *I* for moments of time and *J* for possible worlds (reification):

8.1 DEFINITION OF A MODEL STRUCTURE

The Sextuple $\langle A, I, J, B, F, g \rangle$ is a *model structure*, where *A* is an infinite set of objects or individuals, *I* an infinite set of moments of time, *J* an infinite set of possible worlds, *B* a finite set of basic expressions, *F* a denotation function from *B* into A^* , and *g* an assignment function from variables into A^* . The elements of *J* are in linear order and $I \times J$ is the Cartesian product of *I* and *J*.

The operators of 1.1 (3, 4) are extended to \Box (necessary) and \Diamond (possible):

Syntax:

- (i) If ϕ is a sentence, then $\Box\phi$ is a sentence.
- (j) If ϕ is a sentence, then $\Diamond\phi$ is a sentence.

Semantics:

- (i) If ϕ is a sentence, then $\Box\phi^{\mathcal{M}, i', j', g}$ is 1 if and only if $\phi^{\mathcal{M}, i', j', g}$ is 1 for all $\langle i', j' \rangle \in I \times J$.
- (j) If ϕ is a sentence, then $\Diamond\phi^{\mathcal{M}, i', j', g}$ is 1 if and only if $\phi^{\mathcal{M}, i', j', g}$ is 1 for at least one $\langle i', j' \rangle \in I \times J$.

This construct of possible worlds raises the question: How can truth values characterize content? Let us consider an analogy from technology. On the computer screen, a black and white portrait is composed of two kinds of pixels, black and white, yet the pixel arrangement can be recognized as an individual face, and the more pixels the sharper the image. This is similar in a formal model with propositions which have only two denotations, \emptyset and $\{\emptyset\}$.

Next compare a photo and a movie, both in black and white: they differ in that the pixels in a photo are static, as in the formal model 6.3, while the pixels in the movie change in time and location, as in the formal model structure 8.1. Like models, model structures rely on definitions produced by hand, resembling animation in film.

TV and DBS, in contrast, may create content automatically, one by recording the agent-external reality, the other by monitoring cognition-external data. Furthermore, the move from black and white TV to color TV resembles the move from a model structure to DBS insofar as a color TV has, roughly speaking, three kinds of pixel: R (red), G (green), and B (blue), known as RGB, while DBS has three kinds of core attributes, i.e. *noun*, *adj*, and *verb* (but also a large number of elementary concepts serving as core and continuation values).

TV and DBS differ in that (i) TV is limited to the output modalities of visual and auditory display, while DBS integrates a wide range of modalities in recognition as well as action, and (ii) TV is limited to displaying recorded images and sound, while

DBS spontaneously processes (a) fresh input of raw data directly into content (recognition) and (b) currently activated content directly into raw data output (action), thus satisfying Hume's famous *homunculus* argument⁷ (Hume 1748). The technical basis is a content-addressable memory with a now front, clearing the now front in functional intervals, automatic word form recognition and production, operations for connecting input into content, for activating content by navigation, etc., which are all absent in TV.

9 Semantic Relations of Structure in DBS

The proplets of (i) an elementary proposition and (ii) between the top verb proplets of coordinated propositions (TExer 2.1.6 ff.) are connected by four kinds of relations:

9.1 THE FOUR SEMANTIC RELATIONS OF STRUCTURE

1. subject/predicate
2. object\predicate
3. modifier|modified
4. conjunct—conjunct

The following content is defined as set of proplets. The subject/predicate relation is coded in two directions: (i) between the core feature [noun: [person x]] of *lucy* and the continuation feature [arg: [person x] square] of *find* and (ii) the core feature [verb: find] of *find* and the continuation feature [fnc: find] of *lucy*:

9.2 CONTENT OF *Lucy found a big blue square* . AS PROPLET SET

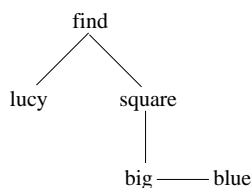
[sur: lucy noun: [person x] cat: snp sem: nm f fnc: find mdr: nc: pc: prn: 14	[sur: verb: find cat: n' a' decl sem: ind past arg: [person x] square mdr: nc: pc: prn: 14	[sur: adj: big cat: adn sem: pad mdd: square mdr: nc: blue pc: prn: 14	[sur: adj: blue cat: adnv sem: pad mdd: mdr: nc: pc: big prn: 14	[sur: noun: square cat: sn sem: sg fnc: find mdr: big nc: pc: prn: 14
---	--	--	--	---

Similarly, the object\predicate relation is coded between (i) the core feature [noun: square] of *square* and the continuation feature [arg: [person x] square] of *find* and (ii) the core feature of [verb: find] of *find* and the continuation feature [fnc: find] of *square*. Correspondingly for *big* and *blue*.

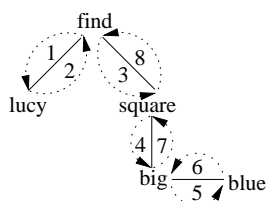
By representing the semantic relations of structure with the operators /, \, |, and —, the content 9.2 may be shown graphically as follows:

9.3 SEMANTIC RELATIONS OF STRUCTURE IN THE CONTENT 9.2

(i) SRG (semantic relations graph)

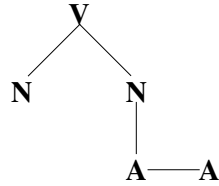


(iii) NAG (numbered arcs graph)



⁷Rejecting a humunculus does not imply rejecting iconicity (FoCL Sect. 3.3).

(ii) signature



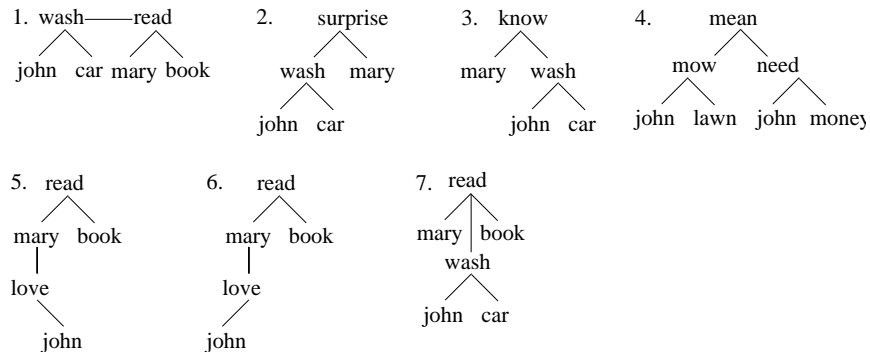
(iv) surface realization

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

The graph is based on the proplets of content 9.2. It characterizes the semantic relations of structure with four views. View (i), called the *semantic relations graph* (SRG), uses the core values *lucy*, *find*, *square*, *big* and *blue* as nodes. View (ii), called the *signature*, uses the core attributes N for noun, V for verb, and A for adj as nodes. View (iii), called the *numbered arcs graph* (NAG), supplements the SRG with numbered arcs which are used in (iv), called the *surface realization*, to show the navigation which 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. The concepts, as the elementary building blocks of Database Semantics, are shown by placeholder values, using English base forms for convenience.

The operators /, \, |, and – are used also in extrapositional semantics:

9.4 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.⁸
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 modifier|modified: 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. For example, 2 requires a psych verb (TExer Sect. 2.5), while 3 requires a mental state verb (TExer Sect. 2.6). Thus, it is impossible linguistically to use the same predicate⁹ for constructing all seven constellations.

Once the semantic structure of a grammatical construction has been figured out graphically, writing the content as a set of proplets, e.g. 9.2, is easy. The subsequent

⁸Verbs which take a clausal subject and a clausal object simultaneously (class 4 in 9.4) include *entail*, *hint*, *imply*, *indicate*, *mean*, *presuppose*, and *suggest*. Thanks to Prof. Kiyong Lee for pointing it out.

⁹The criteria for establishing word classes are diverse (Levin 2009). For analyzing semantic relations of structure, DBS defines verb classes in terms of their possible valency filler (CLaTR Chap. 15).

steps are (a) the hear mode derivation using the lexical proplets provided by automatic word form recognition and (b) the think-speak mode derivation using the content derived in the hear mode (laboratory set-up). For details see TExer.

10 Properties Common to Hear, Think, and Think-Speak Mode Operations

As an agent-based data-driven approach, DBS derivations require three kinds of operations: for (i) the hear mode, (ii) the think mode, and (iii) the think-speak mode. They have in common that they are time-linear and share the following structural properties:

10.1 STRUCTURAL PROPERTIES COMMON TO ALL DBS OPERATIONS

1. DBS operations consist of an antecedent, a connective, and a consequent.
2. The antecedent and the consequent are defined as sets of proplet patterns which are semantically connected by proplet-internal addresses.¹⁰
3. With the exception of inferences, the proplet patterns of the antecedent are the input pattern, while the proplet patterns of the consequent are the output pattern.
4. Inferences apply by matching input to the antecedent (deductive use, CC 3.5.1) or the consequent (abductive use, CC 3.5.2).
5. An operation is activated by content matching the input pattern.
6. Binding the input constants to the variables of the input pattern enables the output pattern to derive the output.
7. The codomain of a variable in a pattern proplet may be restricted by an explicit list of possible values (variable restriction, e.g. CC 5.5.4).

The operations of the hear, think, and think-speak mode of DBS rely on computational pattern matching to apply. Therefore, truly efficient pattern matching is of the essence. Technically, it is based on the data structure of proplets, defined as non-recursive feature structures with ordered attributes. This is the direct opposite of recursive feature structures with unordered attributes (Carpenter 1992), which is standard in today's substitution-driven systems but uniquely ill-suited for pattern matching.

11 DBS Hear Operations

For building complex content in the hear mode of natural language communication, a next word (i.e. a lexical proplet provided by automatic word form recognition) is connected to the current sentence start by one of the following operation kinds:

11.1 THREE KINDS OF DBS HEAR MODE OPERATIONS

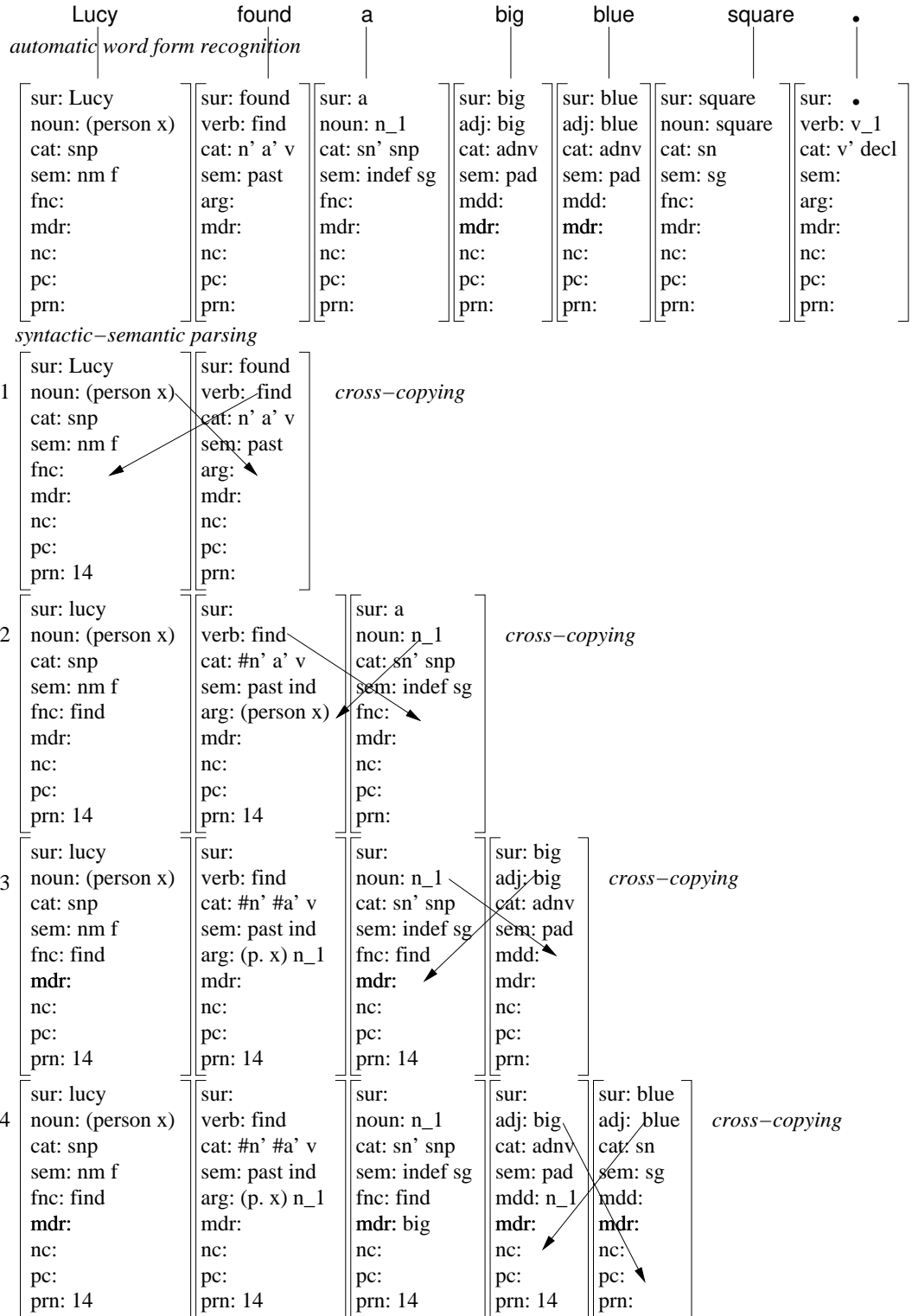
1. cross-copying (connective \times , as in **SBJ** \times **PRD**; TExer 6.3.1, 1)
2. absorption (connective \cup , as in **DET** \cup **CN**; TExer 6.3.1, 51)
3. suspension (connective \sim , as in **ADV** \sim **NOM**; TExer 6.3.1, 32)

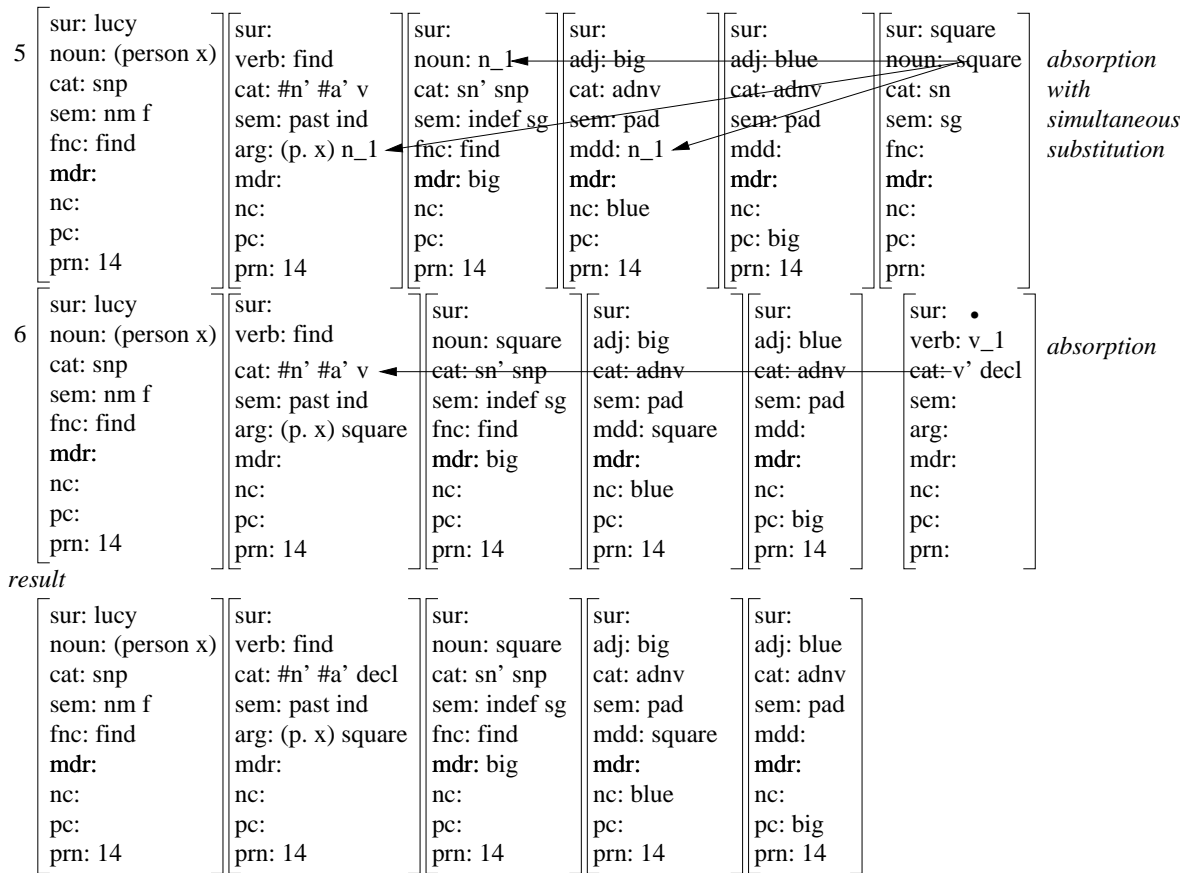
A hear mode operation takes two proplets as input and produces one or two proplets as output.

The following hear mode derivation of the content 9.2 uses operations of the kind (i) cross-copying and (ii) absorption. For (iii) suspension see TExer, Sect. 3.1.

¹⁰With the exception of suspension operations (11.1, 3) in the hear mode.

11.2 TIME-LINEAR SURFACE-COMPOSITIONAL DERIVATION

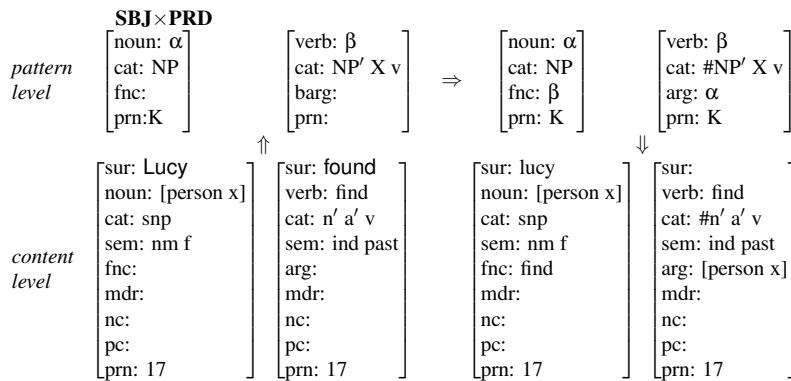




The analysis is (i) *surface compositional* because each lexical item has a concrete SUR value and there are no surfaces without a proplet analysis. The derivation order is (ii) *time-linear*, as shown by the stair-like addition of a next word proplet. The activation and application of operations is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

The computational pattern matching of DBS operations is illustrated by the following application of the operation in line 1 of 11.2:

11.3 CROSS-COPYING *lucy* AND *find* WITH SBJ×PRD (line 1 in 11.2)

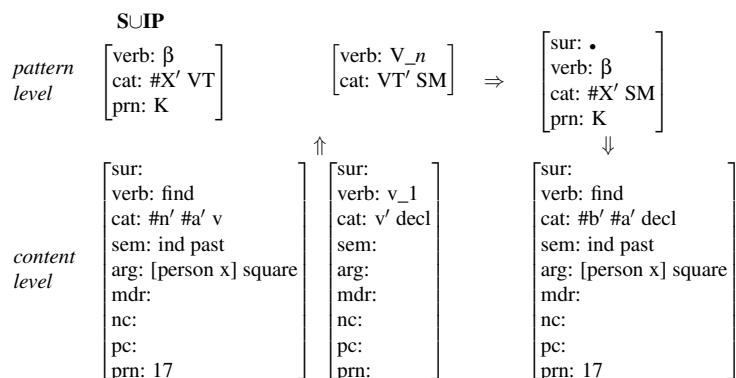


Lexical lookup and syntactic-semantic concatenation are incrementally intertwined:

the lookup of a new next word occurs only after the current next word has been processed into the current sentence start. In each concatenation, the language-dependent sur value provided by lexical lookup is omitted, with the partial exception of names.

While a cross-copying operation like 11.3 produces two output proplets, an absorption operation produces only one. Consider the following example:

11.4 ABSORBING INTERPUNCTUATION INTO *find* WITH SUIP (line 6)



The names of all three kinds of DBS hear mode operations (11.1) have an input1 connective input2 structure, where input2 matches a next word and input1 looks for matching input in the sentence start, i.e. the set of proplets at the current now front.

12 Activation in the DBS Think and Think-Speak Modes

DBS think mode operations are of two kinds, (i) navigation and (ii) inferencing. Navigation operations serve to activate existing content and consist of one input and one output pattern. Inference operations take a given content as input and derive a new content as output; they consist of an open number of input and output patterns. Both kinds have a time-linear derivation order and may be mapped into language-dependent surfaces using the same lex rules as the navigation operations.

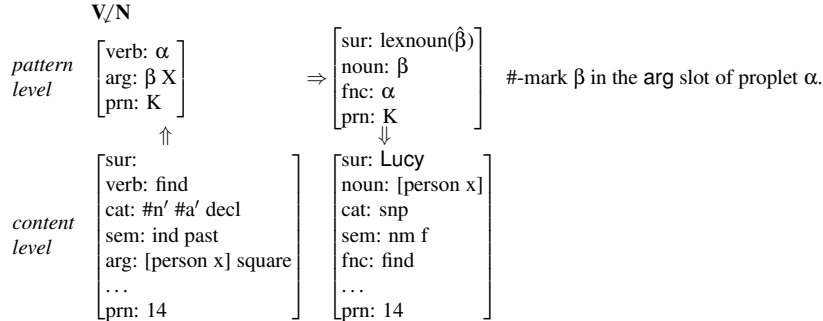
Intrapositionally, navigational DBS think and think-speak mode operations alike activate content by traversing the semantic relations of structure in both directions. Accordingly, the intrapositional operations traversing subject/predicate are $V \setminus N$ and N / V , those traversing object/predicate are $V \setminus N$ and $N \setminus V$, those traversing the modifier/modified relation adnominally are $N \downarrow A$ and $A \uparrow N$ (and similarly for adverbial $V | A$), and those traversing the conjunct-conjunct relation $N - N$ are $N \rightarrow N$ and $N \leftarrow N$ (and similarly for $A - A$ and $V - V$).¹¹

The same holds for extrapositional activations, except that extrapositional coordination is uni-directional in the direction of time and requires an inference for traversal in the anti-temporal direction. In the DBS graph analysis 9.3, which happens to be intrapositional, the dual traversals are shown in the (iii) NAG (numbered arcs graph) and applied in the (iv) surface realization.

The following examples refer to the arc numbers in the NAG of 9.3:

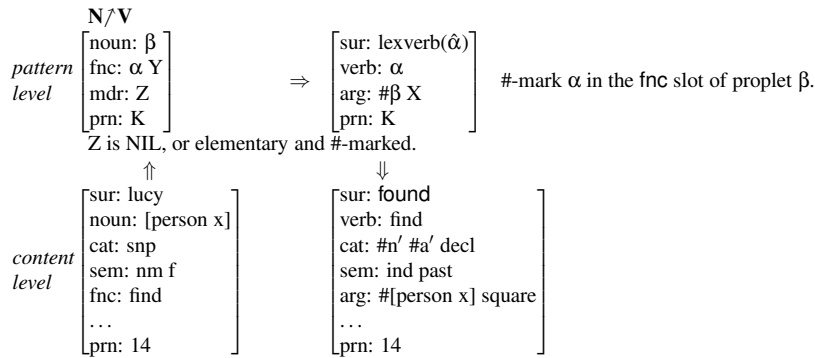
¹¹For a more detailed account see TExer.

12.1 NAVIGATING WITH $V \downarrow N$ FROM *find* TO *Lucy* (arc 1)



In the think mode, the $\text{lexnoun}(\hat{\beta})$ operation in the *sur* slot of the output proplet is switched off, but switched on in the think-speak mode. The variable $\hat{\beta}$ refers to a list which associates each core value with a language-dependent counterpart – with the exception of names, which are realized from a marker in the *sur* slot (CTGR).

12.2 NAVIGATING WITH $N \uparrow V$ FROM *Lucy* BACK TO *find* (arc 2)



Production of the surface *found* by $\text{lexverb}(\hat{\alpha})$ is based on the features [verb: find], [cat: #n' decl], and [sem: ind past] of the output proplet.

13 DBS Inferencing

For DBS inferencing, please see CC, Sects. 3.4-3.6, Chaps. 4 and 5, and Part II.

14 Conclusion

The main difference between the semantics of DBS and PredC (Predicate Calculus) is the agent-based data-driven ontology of DBS vs. the sign-based substitution-driven ontology of PredC. As a consequence, DBS requires on-board interface and memory components, while PredC does not. Also, a sign-based approach has no room for distinct hear and speak modes, while DBS treats them as part of recognition and action.

PredC is perhaps more general and parsimonious, but DBS avoids some longstanding problems of PredC, such as over- (Sect. 2) and undergeneration (Sect. 4), and a dependence of contingent meaning on hand-crafted models (Sects. 6 and 8). Because DBS implements agent-based recognition and action, including the hear (Sect. 11), think, and think-speak modes (Sect. 12), it is more suited for building a talking autonomous robot than systems inherently without on-board interface and memory components.

Bibliography

- Barwise, J., and J. Perry (1983) *Situations and Attitudes*, Cambridge, Mass.: MIT Press
- Brooks, R.A. (1986) "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation* 2.1:14–23
- Carpenter, B. (1992) *The Logic of Typed Feature Structures*, Cambridge: CUP
- 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)
- CTGR = Hausser, R. (2017) "A computational treatment of generalized reference," *Complex Adaptive Systems Modeling*, Vol. 5.1:1–26
- FoCL = Hausser, R. (1999/2001/2014) *Foundations of Computational Linguistics, Human–Computer Communication in Natural Language, 3rd ed.*, pp. 522, Springer
- Frege, G. (1879) *Begriffsschrift. Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle: L. Nebert
- Geach, P. (1972) "A Program for Syntax," in D. Davidson and G. Harman (eds.) *Semantics of Natural Language*, Dordrecht: D. Reidel, pp. 483–497
- Gualterus Burlaeus (1988) *De puritate artis logicae tractatus longior*, Hamburg: Felix Meiner Verlag
- Hausser, R. (1980) "The Place of Pragmatics in Model-Theory," in Groenendijk, J.A.G., T.M.V. Janssen, and M.B.J. Stokhof (eds) (1980) *Formal Methods in the Study of Language*, University of Amsterdam: Mathematical Center Tracts 135
- Hume, D. (1748) *An Enquiry Concerning Human Understanding*, reprinted in Taylor, R. (1974) *The Empiricists: Locke, Berkeley, Hume*, Garden City, New York: Anchor Books, Doubleday
- Kamp, J.A.W. (1980) "A Theory of Truth and Semantic Representation," in J.A.G. Groenendijk et al. (eds.)
- Kamp, J.A.W., and U. Reyle (1993) *From Discourse to Logic*, Parts 1 and 2, Dordrecht: Kluwer
- King, J.C., and K.S. Lewis, "Anaphora," *The Stanford Encyclopedia of Philosophy (Summer 2017 Edition)*, Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/sum2017/entries/anaphora/>
- Knuth, D.E., J.H. Morris, and V.R. Pratt (1977) "Fast Pattern Matching in Strings," *SIAM Journal of Computing* Vol. 6.2:323–350
- Levin, B. (2009) "Where Do Verb Classes Come From?" <http://web.stanford.edu/bclevin/ghent09vclass.pdf>
- Montague, R. (1973) "The Proper Treatment of Quantification in Ordinary English," in J. Hintikka, J. Moravcsik, P. Suppes (eds.) *Approaches to Natural Language*, Dordrecht, Reidel: 221–242
- 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)