Natural Language Production in Database Semantics ⋆

Roland Hausser

Abteilung Computerlinguistik, Universität Erlangen-Nürnberg,
Bismarckstr. 6, 91054 Erlangen, Germany
rrh@linguistik.uni-erlangen.de

Abstract. A theory of natural language production (speaker mode) has to answer the following questions: (i) Where does the content serving as input to language production come from? (ii) In which format is this relatively language-independent content stored, processed, and retrieved? and (iii) How is activated content mapped into well-formed surfaces of a certain natural language? After brief answers to (i) and (ii), this paper concentrates on (iii), illustrating the time-linear method of Database Semantics (DBS) on some of the most notorious grammatical constructions of natural language.

Keywords: Natural language production, speaker mode, center-embedded relative clauses, long distance dependencies, gapping constructions

1 The Coding of Content

A cognitive agent with language has the following sources of content: (i) current non-language recognition, (ii) non-language recognition stored in memory, (iii) inferences which derive new content from current or stored content, and (iv) language interpretation. These contents, regardless of their source, may be processed into blue prints which serve as input to the agent's action components, including natural language production.

In DBS, content is coded, stored, retrieved, and processed in the format of order-free sets of proplets, defined as flat (non-recursive) feature structures. In the following example, the content corresponding to The smart little girl ate an apple. is coded as a set of proplets:

1.1 CODING CONTENT AS A SET OF PROPLETS

noun: girl	adj: smart	adj: little	verb: eat	noun: apple
cat: def sg	sem: pos	sem: pos	sem: past	cat: indef sg
fnc: eat	mdd: girl	pc: smart	arg: girl apple	fnc: eat
mdr: little	nc: little	prn: 23	prn: 23	prn: 23
prn: 23	prn: 23			

Consider the first proplet, *girl*. As a feature structure it consists of attribute value pairs (avp). For better readability and computational efficiency, the avps in a proplet are ordered. The first attribute, noun, is called the core attribute and specifies what corresponds in a content to the parts of speech in natural language. The value of the core attribute is a concept, here girl.

A proplet may be referred to by its core value, written in italics. Thus, example 1.1 consist of the proplets *girl*, *smart*, *little*, *eat*, and *apple*. The proplets of a proposition are held together by the common value of the last attribute, prn (for proposition number, here 23). Using the core and prn value as the primary key, the proplets of a proposition may be retrieved regardless of their storage location as assigned by the database (cf. 2.1) by using that software's retrieval mechanism.

In DBS, there are only three basic kinds of proplets, N (for noun, here *girl* and *apple*), V (for verb, here *eat*), and A (for adjective, here *smart* and *little*. These are connected by seven basic

⋆ This paper benefitted from comments by B. MacWhinney, Carnegie Mellon University; Kiyong Lee, Korea University; Minhaeng Lee, Yonsei University; and J. Handl, T. Proisl, B. Kabashi, and C. Weber, research and teaching associates at Computational Linguistics, Uni Erlangen (CLUE).

intrapropositional semantic relations of structure,[1] listed below and shown graphically as binary relations called elementary signatures:

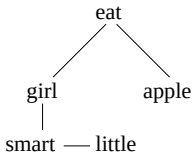
1.2 SEMANTIC RELATIONS OF STRUCTURE AND THEIR ELEMENTARY SIGNATURES

subject verb: N\V
object verb: N/V
adjective noun: A|N
adjective verb: A|V
conjunct conjunct: N—N, V—V, A—A

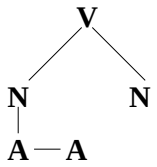
The elementary signatures may be combined to characterize complex contents as graphs. The following two graphs characterizing the content 1.1 are called a *semantic relations graph*, or SRG for short, and a *part of speech signature*, or signature for short:

1.3 REPRESENTING A CONTENT AS TWO KINDS OF GRAPH

(i) *semantic relations graph (SRG)*



(ii) *part of speech signature (signature)*



The order of the letters in an elementary signature like N/V does not indicate any primacy of one part of speech over the other. Instead, the order is motivated graphically: in the case of /, \, and | signatures, the first node is shown lower than the second node in the graph, while in — signatures the first node is shown to the left of the second node in the graph.

The relation between the graphs in 1.3 and the proplet representation 1.1 may be established formally by means of the following schemata, each consisting of two concatenated proplet patterns:

1.4 SCHEMATA INTERPRETING INTER-PROPLET RELATIONS

(i) subject−verb /

noun: α	verb: β
fnc: β X	arg: α

(ii) object−verb \

noun: α	verb: β
fnc: X β	arg: α

(iii) noun−adnominal |

adj: β	noun: α
mdd: α	mdr: β

(iv) verb−adverbial |

adj: β	verb: α
mdd: α	mdr: β

(v) noun coordination —

noun: α	noun: β
nc: β	pc: α

(vi) verb coordination —

verb: α	verb: β
nc: β	pc: α

(vii) adjective coordination —

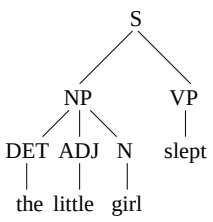
adj: α	adj: β
nc: β	pc: α

The schemata may be applied to the proplets in 1.1 by matching and binding the variables in the patterns to the corresponding constants in the content proplets. In this way, any set of proplets representing a proposition may be mapped automatically into a corresponding DBS graph.

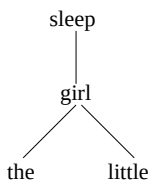
The DBS graphs of content differ from the familiar tree structures of Phrase Structure Grammar (PSG) and Dependency Grammar (DG), for example, in that the lines in a DBS graph have an interpretation which directly characterizes the semantic relations of structure.

1.5 COMPARING THE GRAPH STRUCTURES OF THREE DIFFERENT THEORIES

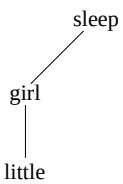
Phrase Structure Grammar



Dependency Grammar



DBS



In Phrase Structure Grammar, the lines specify (i) *dominance* and (ii) *precedence*. Therefore, the grammatical relations of subject verb, object verb, modifier modified, and conjunct conjunct, are not characterized directly, but must be deduced from the dominance and precedence constellations of the nodes. In Dependency Grammar, the lines characterize *dependency* – without characterizing the different roles of the and little, for example. In a DBS graph, in contrast, there are four kinds of lines, /, \, |, and —, each with its own, specific semantic interpretation.

In the DBS hearer mode, lexical proplets representing function words such as the and a(n) are *absorbed* into the associated content word proplets.[2] Therefore, they do not appear in the DBS graphs and must be *precipitated* during language production in the DBS speaker mode.

2 Language Production Based on Navigation

Traditionally, a model of natural language production (speaker mode) has to answer two questions: *What to say?* and *How to say it?* In Database Semantics, the first question is answered in the most general manner as a navigation through the content of a database called Word Bank.

A Word Bank resembles a classic network database (Elmasri and Navathe, 1989) with a column of owner records, here defined in terms of core values, and each owner record associated with a list of member records, here distinguished in terms of their prn values. However, instead of using member and owner *records* we use equivalent member and owner *proplets*:

2.1 STORING THE PROPLETS OF 1.1 IN A WORD BANK

member proplets	*position for new member proplets*	*owner proplets*
	…	…
… [noun: apple, cat: def sg, fnc: grow, prn: 12] [noun: apple, cat: indef sg, fnc: eat, prn: 23]		[core: apple]
	…	…
… [verb: eat, sem: past, arg: John steak, prn: 17] [verb: eat, sem: past, arg: girl apple, prn: 23]		[core: eat]
	…	…
… [noun: girl, cat: indef sg, fnc: see, mdr: tall, prn: 8] [noun: girl, cat: def sg, fnc: eat, mdr: little, prn: 23]		[core: girl]
	…	…
… [adj: little, sem: cmp, mdd: table, prn: 14] [adj: little, sem: pos, pc: smart, prn: 23]		[core: little]
	…	…
… [adj: smart, sem: pos, mdd: girl, nc: rich, prn: 11] [adj: smart, sem: pos, mdd: girl, nc: little, prn: 23]		[core: little]
	…	…

[1] Semantic relations of structure are distinct from semantic relations of meaning, such as *synonymy, antonymy, hypernymy, hyponymy, meronymy,* and *holonymy* as well as *cause–effect*.

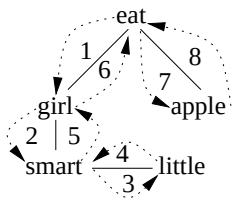
[2] Compare the cat values of *girl* and *apple* in 1.1.

A sequence of member proplets followed by an owner proplet is called a *token line*. The token lines are in the alphabetical order of their core value. Compared to a classic (CODASYL) network database, a Word Bank is highly constrained: the proplets in a token line must all have the same core value (no multiple owners) and are stored in the temporal order of their arrival (reflected by the value of a proplet's prn attribute).

The navigation through the content of a Word Bank is based on the continuation and the prn values of each proplet. For example, the arg value girl and the prn value 23 of the *eat* proplet allow to retrieve (activate, touch, visit, navigate to) the *girl* proplet of the same proposition in the Word Bank, to go from there to the *small* proplet, then on to the *little* proplet, etc. Such a navigation may be represented by a numbered arcs graph, or NAG for short.

2.2 NAG BASED ON THE SRG OF 1.3 AND ASSOCIATED SURFACE REALIZATION

(iii) *numbered arcs graph (NAG)*



(iv) *surface realization*

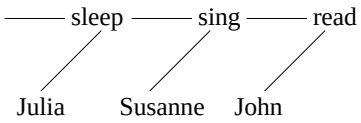
1	2	3	4−5	6	7	8
The	smart	little	girl	ate	the_apple	.

The (iii) NAG is a directed graph. Because it is symmetric, it is graph-theoretically equivalent to the (i) undirected SRG in 1.3. While the NAG shows the time-linear activation order, the (iv) surface realization shows the use of this order for production in a natural language, here English. In language production, the word form surfaces are always realized from the *goal* proplet (goal node) of an arc traversal. The function words and the word order of the natural language in question are accommodated by the possibilities of empty traversals (here 4) and multiple realizations (here 7).

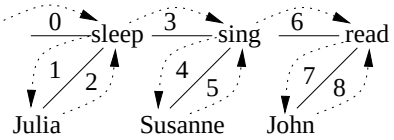
In a DBS content analysis, (i) the semantics relation graph (SRG), (ii) the signature, (iii) the numbered arcs graph (NAG), and (iv) the surface realization are shown together to simultaneously provide four alternative views. This is illustrated by the following analysis of the content corresponding to the extrapropositional coordination Julia sang. Sue slept. John read.:

2.3 REPRESENTING AN EXTRAPROPOSITIONAL COORDINATION

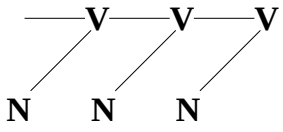
(i) *semantic relations graph (SRG)*



(iii) *numbered arcs graph (NAG)*



(ii) *signature*



(iv) *surface realization*

0−1	2	3−4	5	6−7	8
Julia	slept_.	Susanne	sang_.	John	read_.

In text- or dialog-initial position, the initial — coordination line without a left-hand node serves as the *start line*. Subsequent extrapropositional — lines have a left-hand node, representing the verb of the preceding proposition.

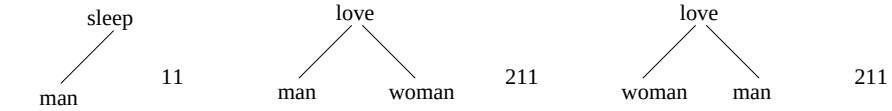
3 Relative Clauses

Can our system as described so far deal with difficult constructions of natural language, and by extension difficult constructions of content? As a case in point, let us compare relative clauses in English and German, including extraposed and center-embedded constructions.

Relative clauses function as sentential (extrapropositional[3]) adnominal modifiers, in contradistinction to adjectival clauses, which function as sentential adverbial modifiers. As a quick introduction to relative clauses in DBS, compare the following DBS graph structures (SRGs) of five simple relative clauses with their corresponding main clauses:

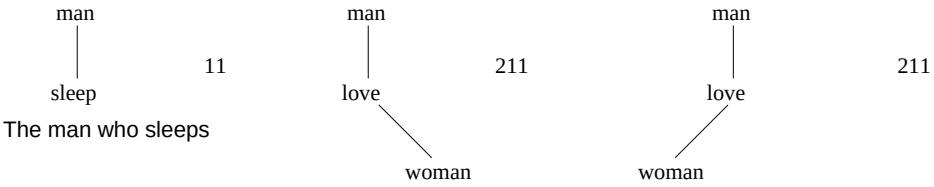
3.1 MAIN CLAUSES AND EQUIVALENT RELATIVE CLAUSES

main clause



The man sleeps. The man loves a woman. A woman loves the man.

relative clause



The man who sleeps

The man who loves a woman

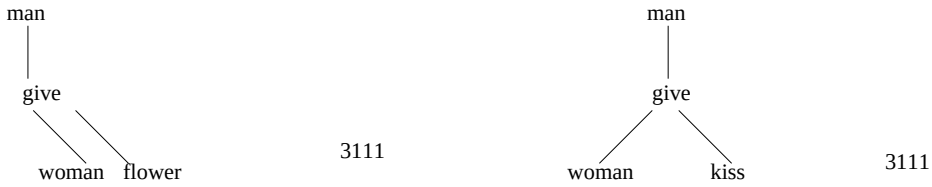
The man whom a woman loves

main clause

The man gives the woman a flower.

The woman gives the man a kiss.

relative clause



The man who gives the woman a flower

The man whom the woman gives a kiss.

The numbers written to the right of each graph are the *degree sequence* (a central notion of graph theory). We see that the degree sequence of a relative clause is the same as that of its main clause.

Relative clauses use their modified ("head noun") as their subject, object, or prepositional argument. For example, in The man who loves a woman (subject gap) the graph shows no subject, just as in The man whom the woman loves (object gap) the graph shows no object. The "shared" noun, i.e., the modified, is graph-theoretically at a minimal distance to the verb of the relative clause, which is no further than the distance between a noun and the verb in the corresponding main clause construction.

Turning to center-embedded relative clauses, we begin with an intuitive structural representation:

3.2 EXAMPLE OF RELATIVE CLAUSE CENTER EMBEDDING

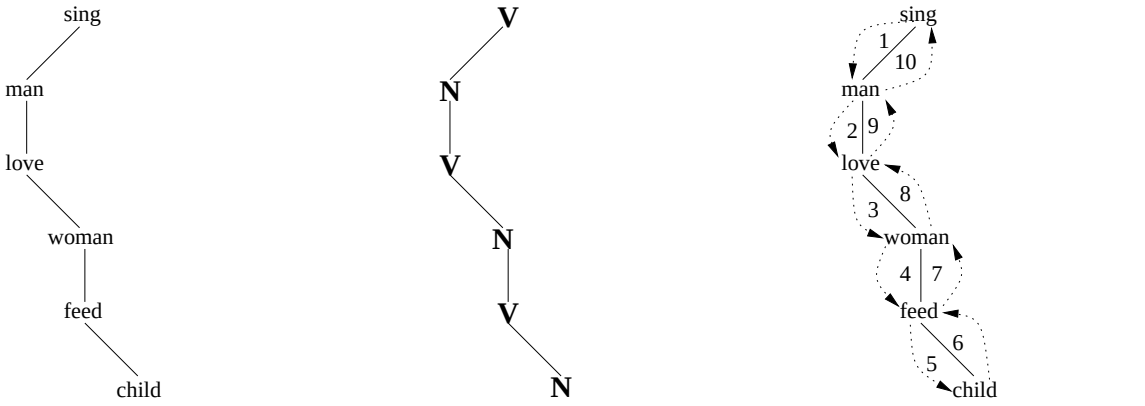
German:	Der Mann					singt
		der die Frau			liebt	
			die das Kind	füttert		
Trans-literation:	the man	who the woman	who the child	feeds	loves	sings

[3] See Sects. 7.4 and 7.5 in (Hausser, 2006) for hearer mode derivations and proplet analyses.

Based on the meaning of the /, \, |, and — lines, the DBS graph analysis of the center embedded relative clause construction 3.2 turns out to be surprisingly simple :

3.3 GRAPH ANALYSIS OF CENTER-EMBEDDED RELATIVE CLAUSES

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



(iv) surface realization (German, center embedded)

1	2	3	4	5	6	7−8	9−10
Der Mann	der	die Frau	die	das Kind	fuettert	liebt	singt .

The DBS graphs (i-iii) have six nodes and the degree sequence 222211. The (ii) signature is given by the native speakers' intuitions about functor-argument and coordination structure. These intuitions are language-independent in the sense that the speakers of different languages can agree where two graph structures are alike and where they differ, for example, because of different lexicalization or different syntactic-semantic coding methods such as the *ergative*.

The German surface realization walks down the left side of the NAG to realize the nouns, then back on the right side to realize the verbs. Double traversals and multiple realizations are evenly spread. Each arc is visited once and the traversal begins and ends with the main verb.

In comparison, consider the English counterpart:

3.4 ENGLISH REALIZATION OF CONTENT 3.3

The man who loves the woman who feeds the child sings.

Because English subclauses have the verb in post-nominative position (in contrast to the verb-final position in German subclauses), the first relative clause may be completed before the second begins. Therefore all nouns and almost all verbs may be realized on the way down the left side of the NAG in 3.3:

3.5 ENGLISH SURFACE REALIZATION OF RELATIVE CLAUSES

surface realization (English, unmarked)

1	2	3	4	5	6−7−8−9−10
The man	who loves	the woman	who feeds	the child	sings .

On the way back up on the right side of the NAG, there is nothing left to do except at the very end (arc 10), when the main verb and the full stop are realized.

Another possibility of English word order is the *extraposition* of a relative clause, as in the following variant of 3.5:

3.6 SURFACE REALIZATION WITH EXTRAPOSED RELATIVE CLAUSE

surface realization (English, extraposed, marked)

1	10	1−2	3	4	5	6−7−8−9−10
The man	sings	who loves	the woman	who feeds	the child	.

The realization of this surface requires a *multiple visit*. After realizing the man in arc 1, the navigation returns to the verb and realizes sings in arc 10. From there, there is no choice but to traverse arc 1 again (multiple visit) and continue with arc 2 to the relative clause. From there the navigation travels down the left side of the graph, realizing the verbs and the nouns. On the way back up, there is nothing left to do except to realize the full stop.

The surface realization 3.6 shows that a consecutive numbering is a sufficient, but not a necessary, condition for satisfying continuity. For example, the combined traversal of arcs 1 and 10 is continuous even though the arc numbers are not numbered consecutively.

From the software side, the mechanism of multiple visits is easily programmed. From the cognitive and the linguistic side, however, multiple visits must be constrained, because otherwise there is no limit on complexity. As a suitable constraint consider the following definition, which applies to connected graphs:

3.7 CONSTRAINT ON MULTIPLE VISITS

A multiple visit is

1. *permitted* if there are still untraversed arcs in the graph,
2. *prohibited* if all arcs in the graph have been traversed.

   (1) and (2) hold for content navigation. (2) may be relaxed by setting limits on the number of traversals a node may have. If there is also language realization, (1) and (2) are modified by the following conditions:

3. *permitted* if function words required by the language have not yet been realized,
4. *prohibited* if there is no value remaining in the current proposition's set of proplets which hasn't already been used exhaustively for realization.

Conditions (1) and (2) are based on the possibility to keep track of how often a node has been traversed. Conditions (3) and (4) are based on the mechanism for mapping content into surfaces.

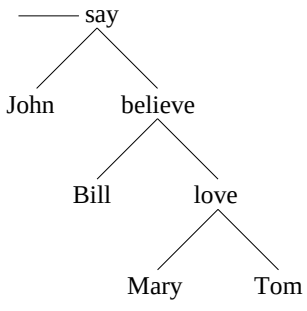
4 Unbounded Dependencies

In the previous section, the same NAG (3.3) was used for realizing the surface (i) of a non-extraposed English relative clause (unmarked case, 3.5) and (ii) of the corresponding extraposed relative clause (marked case, 3.6). This method, based on multiple visits for realizing the marked case, may also be applied to a construction of English known as *unbounded dependency* or *long distance dependency*.

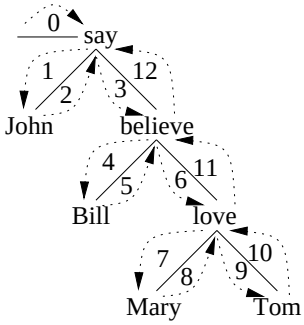
The unmarked case of this construction is an iteration of object sentences, as in the following example:

4.1 DBS GRAPH ANALYSIS FOR John said that Bill believes that Mary loves Tom.

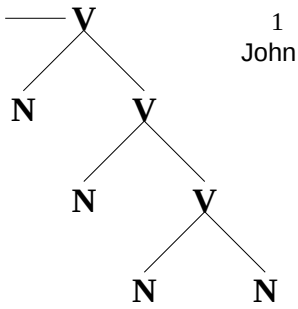
(i) semantic relations graph (SRG)



(iii) numbered arcs graph (NAG)



(ii) signature *(iv) surface realization*



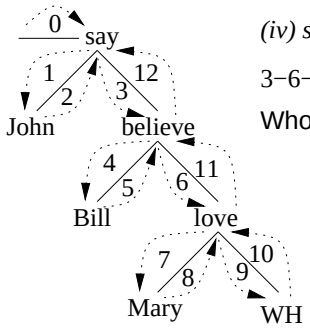
1	2	3	4	5	6	7	8	9	10−11−12
John	said	that	Bill	believes	that	Mary	loves	Tom	.

The iteration may be continued indefinitely, for example, by replacing the last object sentence with that Mary suspects that Susy loves Tom or with that Mary suspects that Susy knows that Lucy loves Tom, etc.

Next consider the corresponding unbounded dependency construction:

4.2 REALIZING Who did John say that Bill believes that Mary loves?

(iii) numbered arcs graph (NAG)



(iv) surface realization

3−6−9	10−11−12	1	2	3	4	5	6	7	8	11−12
Who	did	John	say	that	Bill	believes	that	Mary	loves	?

The (i) semantic relations and (ii) signature graphs of the iterated object sentence construction 4.1 and the corresponding unbounded dependency 4.2 are the same; both have the degree sequence 2221111 (disregarding the start line). The only difference between the respective NAGs is in the object of the final object sentence, i.e., Tom vs. WH. The dependency in 4.2 is between the surface-initial WH and the "underlying" object of the final object sentence; it is unbounded because there is no limit on the number of iterated object sentences.

The order of proplets in the following content representation of 4.2 follows the NAG. Given that proplets are inherently order-free, they could just as well be shown in the alphabetical order of the core values or in the surface order:

4.3 PROPLET REPRESENTATION OF 4.2

verb: say arg: John (believe 7) prn: 6	noun: John fnc: say prn: 6	verb: believe arg: Bill (love 8) fnc: (say 6) prn: 7	noun: Bill fnc: believe prn: 7	verb: love arg: Mary WH fnc: (believe 7) prn: 8			
noun: Mary fnc: love prn: 8	noun: WH fnc: love prn: 8						

The verb *say* has the arguments *John* and *believe*, the verb *believe* has the arguments *Bill* and *love*, and the verb *love* has the arguments *Mary* and WH. Conversely, the object sentence represented by *love* serves as object of the higher clause represented by *believe*, and the object sentence represented by *believe* serves as object of the higher clause represented by *say* (bidirectional pointering).

5 Gapping Constructions

A veritable pinnacle of grammatical complexity is *gapping*. In contrast to sentential modifiers and arguments, which are extrapropositional (cf. Sects. 3, 4), gapping is an intrapropositional construction in which one or more gapping parts share a subject, a verb, or an object with a complete sentence. In addition, there is noun gapping in which several adnominals share a noun.

In common practice, gapping is very rarely used, yet there are strong native speaker intuitions from a wide range of languages[4] confirming that the various forms of gapping exist in their native tongue. This allows only one conclusion: gapping must be a very basic if not primitive construction which closely mirrors the underlying content structure.

The most basic structural distinction between different kinds of gapping in English is whether the filler precedes or follows the gap(s). Subject and verb gapping have in common that the filler comes first. Consider the following examples:

5.1 SUBJECT GAPPING

Bob ate an apple, # walked the dog, and # read the paper.

The *subject* of the complete sentence, Bob, is shared by the following gapping parts, with # marking the gaps.[5]

5.2 VERB GAPPING

Bob ate an apple, Jim # a pear, and Bill # a peach.

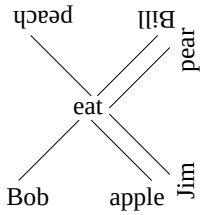
The *verb* of the complete sentence, ate, is shared by the following gapping parts, with # marking the gaps.[6]

Motivated by the presence of and, the gapping analysis in (Hausser, 2006) assumed a coordination structure. This worked well for subject and object gapping, in which a coordination may be defined between the different verbs contained in the gapping parts. But what to do if the gapping parts do not contain any verb because it is the verb that's gapped?

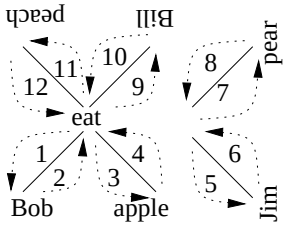
When we finally subjected verb gapping to a DBS graph analysis, the following solution was clear to see:

5.3 VERB GAPPING: Bob ate an apple, Jim a pear, and Bill a peach.

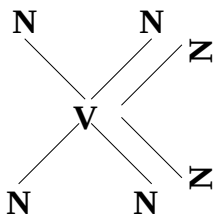
(i) semantic relations graph (SRG)



(iii) numbered arcs graph (NAG)



(ii) signature



(iv) surface realiztion

1	2	3	4−5	6−7	8	9	10−11	12
Bob	ate	the apple	Jim	the pear	and	Bill	the peach	·

[4] They include Bulgarian, Chinese, Czech, English, French, German, Italian, Japanese, Korean, Polish, Romanian, Russian, Spanish, and Tagalog.

[5] In a detailed investigation, T. Proisl found that less than 0.3% of the sentences in the BNC show this construction of subject gapping.

[6] T. Proisl found that less than 0.01% of the sentences in the BNC show this construction of verb gapping.

For graphical reasons, the lines representing the subject verb and the object verb relations in the gapping parts Jim a pear and Bill a peach are oriented properly only when rotated (see orientation of the writing in the semantic relations). The n=7 signature has the degree sequence 6111111. The surface realization uses each arc in the NAG once (no multiple visits), with empty traversals in 4, 6, and 10, and double realizations in 3, 7, and 11.

The graph analysis 5.3 requires a proplet representation different from the coordination-oriented solution of Sect. 8.5 in (Hausser, 2006): here the semantic relations between the gapping constructions and the filler are run via the arg attribute of the lone verb and the fnc attributes of the nouns in the gapping constructions – and not via their nc and pc attributes.

5.4 CONTENT OF VERB GAPPING AS A SET OF PROPLETS

noun: Bob	verb: eat	noun: apple	noun: Jim	noun: pear	noun: Bill	noun: peach
cat: nm	cat: decl	cat: snp	cat: nm	cat: snp	cat: nm	cat: snp
sem: sg	sem: past	sem: indef sg	sem: sg	sem: indef sg	sem: sg	sem: indef sg
fnc: eat	arg: Bob apple	fnc: eat	fnc: eat	fnc: eat	fnc: eat	fnc: eat
prn: 31	Jim pear	prn: 31	prn: 31	prn: 31	prn: 31	prn: 31
	Bill peach					
	prn: 31					

The sharing of the verb is expressed by listing the three subject object pairs in the arg attribute of the verb proplet and the verb in the fnc slot of the nouns (bidirectional pointering). The analysis of verb gapping in terms of functor-argument structure rather than coordination may be applied to all the other forms of gapping [omitted for reasons of space].

Conclusion

This paper presents an agent-oriented approach to language production – in contrast to systems based on such agent-less applications as automatic weather reports, reports on ship locations, train schedules, and the like. The DBS approach requires an agent with a body and external interfaces for recognition and action. These are connected by a database supporting an autonomous control for maintaining the agent in a state of balance vis-à-vis a constantly changing environment.

This overall approach provides language production with a declarative specification of (i) a data structure for representing content, (ii) a database schema for storing and retrieving content, (iii) elementary contents defined as basic recognition and action procedures (lexical semantics), and (iv) complex contents based on semantic relations of structure (compositional semantics).

For our software reconstruction of language production, the agent's action may be confined to the *speaker mode* (producing natural language surfaces from content). Moreover, the *What to say?* part, played by the agent's autonomous control and realized as the selective activation of content for maintaining balance, may be taken temporarily by the linguist selecting contents by hand.

The focus of this paper is a graph-theoretical representation of content, serving as the simplified start structure of the speaker mode (as well as the goal structure of the hearer mode). By defining directed graphs indicating the time-linear activation of content, language production may be analyzed in a conceptually clear and simple format which translates directly into efficiently running code. After introducing our method with elementary intra- and extrapositional constructions, it is tested successfully on relative clause, unbounded dependency, and gapping constructions.

References

Hausser, R. 2006. *A Computational Model of Natural Language Communication*. Berlin Heidelberg New York: Springer.

Elmasri, R., and S.B. Navathe 1989. *Fundamentals of Database Systems*. Redwood City, CA: Benjamin-Cummings