24 Exercises in Linguistic Analysis
DBS Software Design for the Hear and the Speak Mode of a Talking Robot

Preprint June 25, 2017
ROLAND HAUSSER
1989–2011 Professor für Computerlinguistik
Friedrich Alexander Universität Erlangen Nürnberg (em.)

email: rrh@linguistik.uni-erlangen.de
©Roland Hausser, June 25, 2017
Database Semantics (DBS) takes a new look at the analysis of natural language by (i) using a strictly time-linear derivation order (empirical aspect), (ii) providing separate analyses for the hear and the speak mode (functional aspect), and (iii) designing the linguistic analyses as running software for upscaling and automatic verification (methodological aspect). As the ontological foundation and for functional completeness, DBS is designed as the language communication software of a talking robot.

Functional completeness requires a computational model of how communication with natural language works. As a software mechanism, this requires the design of a suitable data structure, algorithm, database schema, and a data-driven self-organization of the operation applications.

Having dealt extensively with functional completeness in CoL, FoCL, NLC, and CLaTR, we turn here to its counterpart, namely completeness of data coverage. This requires the analysis of isolated examples, as in sign-oriented, rule-based (non-statistical) linguistics.

In agent-based DBS, dealing with isolated examples is made possible by adopting the linguistic laboratory set-up: the content derived from a language surface in the hear mode is used by the speak mode as input to be mapped back into the original surface. This ensures that all grammatical properties needed for correct surface production have been taken care of during surface interpretation.

The following linguistic analyses are agent-based in that they connect automatic word form recognition in the hear mode (input) with automatic word form production in the speak mode (output). Also, DBS grammars interpret and produce sequences of sentences (extrapropositional coordination), as in a text \(^2\) and in dialog \(^3\). For using DBS hear and speak grammars as the language communication component of a talking robot, the conditions of the laboratory set-up are replaced by the goal to maintain the agent in a state of balance (action) vis à vis a changing environment (recognition) \(^4\).

A nontrivial sample set which is analyzed explicitly in a linguistically coherent, functionally comprehensive, and computationally efficient framework is rare if not non-existent in today’s linguistics. Each of the 24 analyses in Chaps. \(^2\)-\(^5\) is representative of a whole example class, the elements of which

---

1 CLaTR 4.1.2.
2 NLC Chaps. 11 (hear mode) and 12 (speak mode).
3 CLaTR Chap. 10.
4 CLaTR Chap. 6. References to NLC and CLaTR refer to the preprints of their second edition.
differ in their core values, but share (i) the semantic relations of structure, (ii) the operations of the hear mode, and (iii) the operations of the speak mode.

The data coverage of 24 examples may be considered small, but their choice is linguistically informed: they cover constructions long known among grammarians to be structurally interesting as well as challenging, regardless of the statistical frequency of their use. As an exercise in systematic upscaling, the present fragment may be extended with additional constructions of English or complemented with analogous constructions in other languages. The title of this book is abbreviated with the acronym TExer.

**Abbreviations Referring to Preceding Work**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Work Title</th>
<th>Page Numbers</th>
<th>Publisher</th>
</tr>
</thead>
</table>
2nd Ed. 2017, pp. 363, preprint at lagrammar.net


References are to (preprints of) the latest editions.
# Contents

## 1. Introduction
- 1.1 Hear Mode: Mapping Surfaces into Content ....................................................... 1
- 1.2 Speak Mode: Mapping Content into Surfaces .......................................................... 6
- 1.3 Three Levels of Grammatical Complexity ............................................................... 10
- 1.4 Arc Numbering and Traversal Order ......................................................................... 14
- 1.5 DBS Laboratory Set-Up for Linguistic Analysis ......................................................... 19
- 1.6 The Test List ............................................................................................................. 22

## 2. Obligatory Functor-Arguments
- 2.1 DBS.1: Elementary Subject/Predicate ...................................................................... 33
- 2.2 DBS.2: Phrasal Subject and Phrasal Predicate .......................................................... 39
- 2.3 DBS.3: Elementary Subject/Predicate/Object .............................................................. 48
- 2.4 DBS.4: Phrasal Subject and Phrasal Object ................................................................. 54
- 2.5 DBS.5: Clausal Subject ............................................................................................. 65
- 2.6 DBS.6: Clausal Object ............................................................................................. 75

## 3. Optional Functor-Arguments and Coordination
- 3.1 DBS.7: Elementary Adverbial Modification ............................................................... 85
- 3.2 DBS.8: Phrasal Adverbial Modification ................................................................... 97
- 3.3 DBS.9: Clausal Adnominal Modifier with Subject Gap ............................................. 107
- 3.4 DBS.10: Clausal Adnominal Modifier with Object Gap ............................................ 116
- 3.5 DBS.11: Clausal Adverbial Modification .................................................................. 126
- 3.6 DBS.12: Coordination ............................................................................................. 134

## 4. Valency Positions
- 4.1 DBS.13: Verb Taking a Prepositional Object ......................................................... 147
- 4.2 DBS.14: Verb Taking a Discontinuous Adnominal .................................................. 156
- 4.3 DBS.15: Verb Taking a Discontinuous Bare Preposition ........................................ 164
- 4.4 DBS.16: Verb Taking an Infinitival Object ............................................................... 171
- 4.5 DBS.17: Copula be Taking a Noun ......................................................................... 180
1. Introduction

This chapter introduces the basic method of linguistic analysis in Database Semantics (DBS). Section 1.1 describes the hear mode mapping from language-dependent surfaces into content. Section 1.2 describes the inverse mapping from content into language-dependent surfaces in the speak mode. Section 1.3 illustrates the elementary, phrasal, and clausal levels of grammatical complexity with examples. Section 1.4 relates the selective activation of a content in the think mode to the distinction between a depth first and a breadth first traversal in graph theory. Section 1.5 presents the linguistic laboratory set-up of DBS. Section 1.6 explains the choice and the ordering of the 24 examples.

1.1 Hear Mode: Mapping Surfaces into Content

An agent-based theory of computational linguistics needs to have (i) a data structure for coding content, (ii) an algorithm to map surfaces into content in the hear mode and to map content into surfaces in the speak mode, and (iii) a database schema for the storage and retrieval of content. The data structure of DBS is non-recursive feature structures with ordered attributes called proplets.

1.1.1 Examples of Lexical Proplets

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: (person x)</td>
<td>verb: buy</td>
<td>noun: n_1</td>
<td>adj: new</td>
<td>noun: car</td>
<td>verb: v_1</td>
</tr>
<tr>
<td>cat: snp</td>
<td>cat: n’ a’ v</td>
<td>cat: sn’ snp</td>
<td>cat: adn</td>
<td>cat: sn</td>
<td>cat:’ decl</td>
</tr>
<tr>
<td>sem: nm m</td>
<td>sem: past</td>
<td>sem: indef sg</td>
<td>sem: pad</td>
<td>sem: sg</td>
<td></td>
</tr>
<tr>
<td>fnc:</td>
<td>arg: mdr</td>
<td>fnc: mdr</td>
<td>mdd:</td>
<td>fnc: mdr</td>
<td></td>
</tr>
<tr>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td></td>
</tr>
<tr>
<td>nc:</td>
<td>nc:</td>
<td>nc:</td>
<td>nc:</td>
<td>nc:</td>
<td></td>
</tr>
<tr>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
<td></td>
</tr>
<tr>
<td>prn:</td>
<td>prn</td>
<td>prn</td>
<td>prn</td>
<td>prn</td>
<td></td>
</tr>
</tbody>
</table>

The second attribute from the top is called the core attribute, e.g. noun, verb, or adj, and its value is called the core value.

Lexical proplets are concatenated into a content by copying values. Consider the following content based on the lexical proplets of 1.1.1:
1.1.2 CONTENT OF John bought a new car.

The semantic relation of subject/predicate is coded by copying the core value (person x) of John into the first arg slot of buy and the core value of buy into the fnc slot of John. The semantic relation of object/predicate is coded by copying the core value of car into the second arg slot of buy and the core value of buy into the fnc slot car, etc.

This form of concatenation is established by the following surface compositional, strictly time-linear hear mode derivation:

1.1.3 TIME-LINEAR HEAR MODE DERIVATION OF A CONTENT

**unanalyzed surface**

<table>
<thead>
<tr>
<th>John</th>
<th>bought</th>
<th>a</th>
<th>new</th>
<th>car</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: (person x)</td>
<td>verb: buy</td>
<td>noun: n_1</td>
<td>adj: new</td>
<td>noun: car</td>
<td>verb: n_1</td>
</tr>
<tr>
<td>sem: nm m</td>
<td>sem: past</td>
<td>sem: indef sg</td>
<td>sem: pad</td>
<td>sem: sg</td>
<td></td>
</tr>
<tr>
<td>fnc: arg</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mdr:</td>
<td>nc:</td>
<td>nc:</td>
<td>nc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prn: 1</td>
<td>prn:</td>
<td>prn:</td>
<td>prn:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>syntactic-semantic parsing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1

| sur: John | sur: bought | cross-copying |
| noun: (person x) | verb: buy | cat: n’ a’ v |
| cat: snp | mdr: | sem: past |
| sem: nm m | nc: | pc: |
| fnc: arg | prn: 1 |

2

| sur: john | sur: bought | cross-copying |
| noun: (person x) | verb: buy | cat: n’ a’ v |
| cat: snp | mdr: | sem: past |
| sem: nm m | nc: | pc: |
| fnc: buy | prn: 1 |

| mdr: | nc: | prn: |
| pc: | prn: |

| mdr: | nc: | prn: |
| pc: | prn: |
In addition to cross-copying, hear mode derivations use absorption, with and without simultaneous substitution, as shown in lines 5 and 6, respectively. Also, if a next word proplet cannot find a suitable proplet to connect with, there is a suspension until a proper candidate arrives.

Cross-copying, absorption, and suspension are performed by operations
which are binary in the sense that they always connect two proplets. This is reflected by the names of hear mode operations: they consist of (i) a term characterizing the first input proplet, (ii) a connective which indicates the kind of operation, and (iii) a term characterizing the second input proplet. Cross-copying is indicated by the connective $\times$, as in $\text{SBJ} \times \text{PRD}$ (2.1.2), absorption by $\cup$, as in $\text{S} \cup \text{IP}$ (2.1.3), and suspension by $\sim$, as in $\text{subclause} \sim \text{SBJ}$ (2.5.3).

The following application of the hear mode operation $\text{SBJ} \times \text{PRD}$ shows the cross-copying in line 1 of 1.1.3:

1.1.4 EXAMPLE OF APPLYING A HEAR MODE OPERATION

The operation name is followed by its number, here (h1), in the hear grammar 6.3.1, which lists for each operation the reference numbers of all its applications.

The pattern level shows the operation. The content level shows the input and the output. The proplets at the content level relate to the pattern proplets by pattern matching. A pattern proplet matches a content proplet if (i) the attributes of the pattern proplet are a sublist of the attributes of the content proplet and (ii) the values of the pattern proplet match corresponding values of the content proplet. In addition to variable-constant matching there is matching based on the type-token relation.

In the hear mode, the proplets are provided by automatic word form recognition. They are stored and connected in the agent’s memory, which is realized as a computational database called word bank. A word bank has a two-

---

1 In the output of a hear mode operation, the sur values are deleted. A partial exception are proplets of the sign kind name, which retain the surface as a marker, written in lower case and default font instead of helvetica.
dimensional database schema. Horizontally, proplets with the same core value are stored in a *token line* in the order of their arrival. Vertically, the token lines are in the alphabetical order induced by their core value (in the case of names also in accordance with their sur marker). Consider how the content derived in 1.1.3 is stored in a previously empty word bank:

### 1.1.5 DATABASE SCHEMA OF A CONTENT-ADDRESSABLE WORD BANK

<table>
<thead>
<tr>
<th>member proplets</th>
<th>now front</th>
<th>owner values</th>
</tr>
</thead>
<tbody>
<tr>
<td>sur:</td>
<td>buy</td>
<td></td>
</tr>
<tr>
<td>verb: buy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cat: #n' a' v</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sem: past</td>
<td></td>
<td></td>
</tr>
<tr>
<td>... arg: (person x) car</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mdr:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prn: 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[sur: ]</td>
<td>car</td>
<td></td>
</tr>
<tr>
<td>noun: car</td>
<td></td>
<td></td>
</tr>
<tr>
<td>car: snp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sem: indef sg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>... fnc: buy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mdr: new</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prn: 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[sur: john</td>
<td>john</td>
<td></td>
</tr>
<tr>
<td>noun: (person x)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cat: snp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sem: nm m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>... fnc: buy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mdr:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prn: 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[sur: new</td>
<td>new</td>
<td></td>
</tr>
<tr>
<td>adj: new</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cat: adn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sem: pad</td>
<td></td>
<td></td>
</tr>
<tr>
<td>... mdd: car</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mdr:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prn: 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Because the semantic relations between proplets are coded by address, they are order-free (CLaTR 3.2.8) in the sense that the semantic relations between

---

2 The two-dimensional structure of a word bank resembles a classic network database with member and owner records (Elmasri and Navathe 2010).
them are preserved regardless of where the proplets are stored. A word bank is content-addressable (Bachman 1973) because storage and retrieval do not require a separate index (inverted file, Zobel and Moffat 2006), but rely entirely on the database schema. The length of a token line depends on the amount of content in the word bank. The length of the owner column depends on the number of core values in the system.

A hear mode operation and the agent’s word bank are integrated as follows: (i) automatic word form recognition stores a next word as a lexical proplet at the now front in the token line of its core value; (ii) this activates all operations (provided by memory) which match it with their second input pattern; (iii) activated operations look for proplets at the now front which match their first pattern and apply.

To make room for the next input to be concatenated, the now front must be cleared in regular intervals. For this, the proplets currently stored at the now front are pushed to the left, as in a pushdown automaton, – or, equivalently, the now front and the column of owner values are moved to the right into free memory space, leaving the proplets to be cleared behind in what is turning into an area of member proplets, like sediment (loom-like clearance).

This mechanism creates a faithful record of the time-linear arrival order of what were - prior to their being connected - lexical next word proplets in their respective token line. The only way to correct content stored in a word bank is by adding a comment at the current now front, like a diary entry. The relation between the comment and the content to be corrected is solely by address, thus leaving the sediment completely untouched.

1.2 Speak Mode: Mapping Content into Surfaces

Language-dependent surfaces play a role only in the automatic word form recognition of the hear mode and the automatic word form production of the speak mode. All other cognitive operations of the DBS agent work with proplets which are independent of surfaces. Language-specific properties regarding agreement, valency structure, and word order are coded into the proplets and the operations of the hear, the think, and the speak mode, making cognition independent of surfaces.

The think mode takes proplets stored in the word bank as input and consists of two parts, (i) selective activation and (ii) inferencing. Selective activation is based on navigating along the semantic relations coded by address between proplets. Inferencing derives new content from activated proplets. An initial activation is provided by an external or internal stimulus.
In the DBS analysis of a complex content, the semantic relations between the proplets are of four kinds, namely subject/predicate, object\predicate, modifier|modified, and conjunct—conjunct. The first three constitute the traditional notion of functor-argument, while the last is coordination. The first two are obligatory, while the latter are optional.

A semantic relation of structure is defined as a binary relation between two proplets. Such a relation may be presented in static and in dynamic form. These in turn may be shown (i) in graph and (ii) in linear notation. The four variants follow a simple logic.

In static form, the basic semantic relations of structure are written as (i) a letter representing the start proplet, one of (ii) the connectives /, \, |, and –, and (iii) a letter representing the goal proplet:

1.2.1 Static presentation of the four semantic relations

<table>
<thead>
<tr>
<th>subject/predicate</th>
<th>object/predicate</th>
<th>modifier</th>
<th>modified</th>
<th>conjunct—conjunct</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph notation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(V) (\rightarrow) (N)</td>
<td>(V) (\rightarrow) (N)</td>
<td>(N) (\rightarrow) (A)</td>
<td>(N) (\rightarrow) (N)</td>
<td></td>
</tr>
<tr>
<td>linear notation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(N/V)</td>
<td>(N/V)</td>
<td>(A/N)</td>
<td>(N–N)</td>
<td></td>
</tr>
</tbody>
</table>

If there is a lower node in a binary graph, it precedes in linear notation.

In dynamic form, the direction of traversing a relation is indicated by numbered arcs in the graph and by extending the slashes into arrows in linear notation:

1.2.2 Dynamic presentation of the four semantic relations

<table>
<thead>
<tr>
<th>subject/predicate</th>
<th>object/predicate</th>
<th>modifier</th>
<th>modified</th>
<th>conjunct—conjunct</th>
</tr>
</thead>
<tbody>
<tr>
<td>graph notation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(V) (\rightarrow) (N)</td>
<td>(V) (\rightarrow) (N)</td>
<td>(N) (\rightarrow) (A)</td>
<td>(N) (\rightarrow) (N)</td>
<td></td>
</tr>
<tr>
<td>linear notation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(N/V)</td>
<td>(N/V)</td>
<td>(A/N)</td>
<td>(N–N)</td>
<td></td>
</tr>
</tbody>
</table>

In graphical functor-argument, forward navigation is shown as downward and backward navigation as upward, but in coordination as left-right and right-left. In linear notation, the start node uniformly precedes the goal node.
The semantic relations of structure in complex contents are analyzed as complex graphs. To bring out different aspects of the linguistic analysis underlying the speak mode, DBS uses four different views. For example, the content \([1.1.2]\) has the following canonical DBS graph analysis:

### 1.2.3 Graph Analysis underlying the Production of [1.1.2]

(i) SRG (semantic relations graph)

(ii) signature

(iii) NAG (numbered arcs graph)

(iv) surface realization

The (i) SRG (semantic relations graph) connects the core values of the proplets in [1.1.2], while the (ii) signature connects the core attributes. These two graphs provide static views on the semantic relations of structure. The (iii) NAG (numbered arcs graph) and the (iv) surface realization, in contrast, show a dynamic (navigational) view on the content.

The (iv) surface realization consists of three lines. They show for each navigation step (a) the number of the arc traversed, (b) the surface produced, and (c) the nature of the transition. For example, the traversal of arc 1 produces the surface \(\text{John}\) using the downward navigation \(V/N\), while the traversal of arc 2 produces the surface \(\text{bought}\) using the upward navigation \(N/V\), etc. A transition sequence, e.g., \(V/N\ N/V\ V/N\ A\ A\ N\ N/V\ \) must obey the continuity condition (NLC 3.6.5), i.e., an operation \(AxB\) may be directly followed by an operation \(BzC\) only if the output of \(AxB\) is matched by the antecedent of \(BzC\), where \(A, B, C\) are patterns and \(y, z\) are connectives.

The operations of the DBS speak mode resemble hear mode operations \([1.1.4]\) in that they are defined as proplet patterns. They differ in that speak

---

3 CLaTR Sect. 7.1 compares DBS graphs with the trees of phrase structure and dependency grammar.
4 The surface realization in a DBS graph analysis uses, e.g., \(V/N\) instead of \(V_/N\) for simplicity and because the direction is specified by the associated numbered arc.
mode operations have only one input and one output pattern. Technically, 
speak mode operations are think mode operations with a language-dependent 
lexicalization rule in the $\text{sur}$ slot of the output pattern.  

The following example shows the downward traversal from the predicate to 
the subject via arc 1 in the NAG of 1.2.3.

### 1.2.4 Downward Traversal with a Speak Mode Operation

\[
\begin{align*}
\text{V\slash N (4)} & \quad \Rightarrow \quad \text{sur}: \text{lexnoun(}$\beta$\text{)} \\
\text{pattern level} & \quad \text{noun}: \beta \\
\text{arg: } \alpha & \quad \text{fnc: } \alpha \\
\text{prn: K} & \\
\uparrow & \\
\text{sur: John} & \quad \text{noun: (person x)} \\
\text{verb: buy} & \quad \text{cat: snp} \\
\text{cat: #n' #a' decl} & \quad \text{sem: nm m} \\
\text{sem: past} & \quad \text{fnc: buy} \\
\text{arg: (person x) car} & \quad \text{mdr:} \\
\text{nc:} & \quad \text{pc:} \\
\text{prn: 1} & \\
\downarrow & \\
\text{sur: bought} & \quad \text{verb: buy} \\
\text{cat: #n' #a' decl} & \quad \text{sem: past} \\
\text{sem: nm m} & \quad \text{arg: (person x) car} \\
\text{fnc: buy} & \quad \text{mdr:} \\
\text{nc:} & \quad \text{pc:} \\
\text{prn: 1} & \\
\end{align*}
\]

The operation name, here $\text{V\slash N}$, is followed by the operation number, here (s1). 
The latter refers to the DBS speak grammar 6.5.1 in which each operation lists 
the reference numbers for all its concrete applications, e.g. 1.2.4.

Next consider the operation $\text{N\slash V}$ as it applies in the corresponding upward 
traversal of arc 2 in 1.2.3.

### 1.2.5 Upward Traversal with a Speak Mode Operation

\[
\begin{align*}
\text{N\slash V (5)} & \quad \Rightarrow \quad \text{sur}: \text{lexverb(}$\alpha$\text{)} \\
\text{pattern level} & \quad \text{verb: } \alpha \\
\text{nun: } \beta & \quad \text{arg: } \beta \ X \\
\text{fnc: } \alpha & \quad \text{prn: K} \\
\text{mdr: } Z & \\
\downarrow & \\
\text{sur: John} & \quad \text{noun: (person x)} \\
\text{verb: buy} & \quad \text{cat: snp} \\
\text{cat: #n' #a' decl} & \quad \text{sem: nm m} \\
\text{sem: past} & \quad \text{fnc: buy} \\
\text{arg: (person x) car} & \quad \text{mdr:} \\
\text{nc:} & \quad \text{pc:} \\
\text{prn: 1} & \\
\end{align*}
\]

\[Z\] is NIL, or elementary and #-marked.

---

5 In inferencing (CLaTR Chap. 5), not only the consequent but also the antecedent may be realized in 
the form of language surfaces.
The argument $\hat{\alpha}$ (hat alpha) of lexverb refers to the language-dependent variant of the core value, e.g., German *kaufen* for *buy*. Here, lexverb uses the core value *buy* and the *sem* value *past* to realize *bought*.

### 1.3 Three Levels of Grammatical Complexity

In natural language, the four semantic relations represented by $/$, $\\backslash$, $\mid$, and $-$ apply at three levels of grammatical complexity, called elementary, phrasal, and clausal. Consider the DBS graph analysis of the short text *The heavy old car hit a beautiful tree. The car had been speeding. A farmer gave the driver a lift.*

#### 1.3.1 Clausal, Elementary, and Phrasal Relations

The semantic relations are defined between content proplets, represented by their core value. In other words, the graph is an SRG (1.2.3).

Elementary and phrasal relations are intrapropositional, while clausal relations are extrapropositional. At the clausal level, the predicates of the three propositions are connected by extrapropositional coordination as the conjuncts *hit*–*speed*–*give*. The predicates form elementary subject/predicate relations with *car*, *car*, and *farmer*, respectively, and elementary object/predicate relations with *tree*, *driver*, and *lift*. The first subject is in the adnominal modifier/modified relation *heavy*|*car* at the phrasal level; the modifier in turn forms the conjunction *heavy*–*old* (intrapropositional coordination). Also, the adnominal *beautiful* is in a modifier/modified relation with the object *tree*.

Next consider word order variation for a given content as an instance of paraphrase (FoCL 4.4.6). A classic example is the active-passive alternation illustrated by the following example using English:

---

6 Whether an expression in the active voice and its passive counterpart have the same content or not has been discussed controversially in the literature. Originally generated from the same “deep structure” by a transformation, the meaning equivalence was later cast into doubt based on differences in quantifier scope. For example, *Everyone in this room speaks at least two languages* (active) and *At least two languages are spoken by everyone in this room* (passive) seem to correspond to the alternative orders of quantifiers postulated by truth-conditional semantics (NLC 6.4.4). The elimination of quantifiers in DBS allows us to treat the literal meaning of active and passive as equivalent, moving the alleged scope ambiguity into the inferencing of pragmatics, where it belongs.
1.3.2 ALTERNATION BETWEEN ACTIVE AND PASSIVE IN ENGLISH

(i) SRG (semantic relations graph)

```
read

john  book
```

(ii) signature

```
V

N  N
```

(iii) NAG (numbered arcs graph)

```
read

john  book
```

(iv) surface realization

```
V/N  N/V  V/N  N/V

b. 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, while the passive traverses them in the order 3, 4, 1, 2.

The content common to the speak mode derivation of the active and the passive variant in 1.3.2 is based on the subject/predicate and object/predicate relations, and defined as follows:

1.3.3 CONTENT COMMON TO THE ACTIVE AND THE PASSIVE VARIANT

```
sur: john
noun: (person x)
cat: snp
sem: nm m
fnc: read
mnr:
nc:
pc:
prn: 2

sur: verb: read
cat: #n' #a' decl
sem: past
arg: (person x) book
mnr:
nc:
pc:
prn: 2

sur: noun: book
cat: snp
sem: indef sg
fnc: read
mnr:
nc:
pc:
prn: 2
```

The difference between John read a book (surface realization a) and A book was read by John (surface realization b) is accommodated by the lexicalization rules (NLC Sect. 12.4–12.6) embedded into the sur slot of the goal pattern of the navigation operations.

Another instance of paraphrase is the word order alternation of the indirect and the direct object in English, as The man gave the child an apple vs. The man gave an apple to the child. Consider the following DBS graph analysis underlying the speak mode realization of both surfaces:

It seems to hold in general that expressions with a symmetric alternation like My home is my castle and My castle is my home are based on the same semantic relations of structure, but have nevertheless different pragmatic implications in natural language.
1.3.4 Surface Alternation in a Three-Place Verb Proposition

(i) SRG (semantic relations graph)

```
  give
  /   \
man-|-child
      |   apple
```

(ii) signature

```
| V | N | N |
```

(iii) NAG (numbered arcs graph)

```
  give
  /   \
man-|-child
      |   apple
```

(iv) surface realizations

a. 1 2 3 4 5 6
   The_man gave the_child an_apple

b. 1 2 5 6 3 4
   The_man gave an_apple to_the_child

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 1.3.2 and 1.3.4 are within the English language, there are similar kinds of paraphrase also between languages. As an example, consider a content common to corresponding surfaces in English and German:

1.3.5 Content of Peter has read a book.

```
| sur: peter |
| noun: (person x) |
| cat: snp |
| sem: nm m |
| fnc: read |
| prn: 79 |

| sur: verb: read |
| cat: #n' #a' decl |
| sem: perf |
| arg: (person x) book |
| prn: 79 |

| sur: noun: book |
| cat: snp |
| sem: def sg |
| fnc: read |
| prn: 79 |
```

Apart from the different word form surfaces of English Peter has read the book and German Peter hat das Buch gelesen, there is a difference in word order. This may be shown with the following DBS graph 1.3.6 underlying the speak mode.

The SRG, the signature, and the NAG are the same for English and German because these graphs characterize the semantic relations in a content, independently of the language-dependent surface realization. The word order difference between the two languages is provided by the language-dependent lexicalization rules, which segment the two surfaces differently in the traversal of different arcs:
1.3.6 WORD ORDER DIFFERENCE BETWEEN ENGLISH AND GERMAN

In English, the complex verb form `has read` 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.

Finally let us turn to a semantic similarity between contents which differs from a proper paraphrase because it is incomplete. For example, the propositions represented by *The little girl slept* and *Fido snored* may be connected extrapropositionally by either the horizontal conjunct—conjunct or the vertical modifier|modified relation:

1.3.7 CLAUSAL CONNECTION BY COORDINATION VS. MODIFICATION

As contents, the constructions are intuitively similar, but they differ syntactically. The extrapropositional alternation is structurally possible because the conjunct—conjunct and modifier|modified relations are both optional.

The variety of extrapropositional 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). Consider the following examples:

---

7 For a comparison of adnominal subclause word order in English and German see CLaTR Sect. 9.3.
1.3.8 Relating two transitive verbs extrapropositionally

1. wash—read  
   john car mary book

2. surprise  
   wash mary

3. know  
   mary wash

4. mean  
   mow need

5. read  
   mary book

6. read  
   mary book

7. read  
   mary 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 (Sect. 2.5), while 3 requires a mental state verb (Sect. 2.6). Thus, it is impossible to use the same predicate for constructing all seven constellations.

1.4 Arc Numbering and Traversal Order

The traversal of a DBS graph for selective activation and surface realization in the speak mode is highly constrained. First, the nodes are traversed in accordance with the Continuity Condition (NLC 3.6.5). Second, within the DBS laboratory set-up (Sect. 1.5), a speak mode derivation must realize the same surface as interpreted by the corresponding hear mode derivation. Third, the overall structure of the graph is determined empirically by the semantic properties of the input content, derived in the hear mode.

As a consequence, there is no choice between different traversals, such as between a depth first vs. breadth first traversal in graph theory. The only aspect open to choice is the arc numbering in a NAG, which may simulate a depth first or a breadth first traversal.
For example, the following two NAGs differ only in their arc numbers: the ones on the left are seemingly depth first, the ones on the right seemingly breadth first.

1.4.1 Example 1: Arc Numbering Depth First vs. Breadth First

Depth first:

**NAG (numbered arcs graph)**

(iii) NAG (numbered arcs graph)

(iii) NAG (numbered arcs graph)

The different arc numbers result in different traversal numbers in the top line of the **surface realizations**: each number specifies for each associated transition directly underneath in the bottom line in which arc of the NAG the transition takes place. Except for the numbering, the NAG and the surface realization on the left are the same as those on the right. Thus, there is a choice of how to number the arcs, but no choice of how to traverse the NAG for surface production in the speak mode, at least not in the DBS laboratory set-up (Sect. 1.5).

The arc numbering on the left of 1.4.1 is linguistically preferable because it produces a consecutive numbering in the surface realization. There are, however, constructions of natural language for which a depth first arc numbering does not fit.

Consider the following example of predicate gapping (Sect. 5.3).

8 Verbs which take a clausal subject and a clausal object simultaneously (class 4 in 1.3.8) include entail, hint, imply, indicate, mean, presuppose, and suggest. Thanks to Prof. Kiyong Lee for pointing it out.

9 The 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 fillers (CLaTR Chap. 15).

10 The arc numbering must be motivated independently from the production order in the natural language surface, here graph-theoretically as depth first vs. breadth first. If the arc numbering were to always follow the surface realization, the distinctions between paraphrases such as active and passive could not be based on alternative traversals of the same NAG. Also, a consecutive arc numbering following the surface order is impossible in constructions with multiple traversals of the same node as in 1.4.3, 1.4.6, 1.4.7, and 1.5.3.
1.4.2 Example 2: Arc numbering Depth first vs. Breadth first

Depth first:

(iii) NAG (numbered arcs graph)

```
   1 2
bob
   3 4
    5 6 7 8
jim
   9 10
    11 12
   peach

buy
```

Breadth first:

(iii) NAG (numbered arcs graph)

```
   1 2
bob
   3 4
    5 6 7 8
jim
   9 10
    11 12
   pear

buy
```

(iv) surface realization

```
1 2 3 4 5 6
Bob bought an_apple Jim
V/N N/V N/V N/V N/V
9 10 5 6 7 8
a_pear and_Bill a_peach .
V/N N/V N/V N/V V/N V/N
```

Again, the semantic relations between the proplets of the content and the order of the word form surfaces are given empirically. The only aspect of the analysis open to choice is the numbering of the arcs in the two NAGs, i.e. between depth first and breadth first. The choice is clear: in contradistinction to 1.4.1, the content structure is breadth first, as shown on the right by the consecutive arc numbering in the surface realization.

Breadth first arc numbering is also used with certain adverbial modifiers (3.2.11, 4.1.10, 4.2.8).

If there is more than one numbering scheme, i.e. breadth first for gapping and depth first for most of the rest, there arises the question of how they should be combined. Consider the following general cases:

1.4.3 Two cases of combining breadth first and depth first

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>clausal coordination</td>
<td>clausal functor–argument</td>
</tr>
<tr>
<td>( \text{df} )</td>
<td>( \text{bf} )</td>
</tr>
<tr>
<td>proposition</td>
<td>proposition</td>
</tr>
</tbody>
</table>

These considerations apply to the unmarked constructions of a language, for example, the active voice. The non-consecutive numbering of the corresponding passive (1.3.2) characterizes it as marked (CLaTR Sect. 9.5). The same holds for the variants \( \text{a} \) and \( \text{b} \) in (1.3.3) and for (1.5.3) (marked) and its declarative counterpart (unmarked).
Case 1 shows three conjuncts of an extrapropositional coordination, as in a text. The middle conjunct is a breadth first gapping construction, marked bf, while the first and the third conjuncts are depth first, marked df.

Case 2 shows the embedding of a breadth first gapping into a depth first matrix. In a breadth first construction the entrance predicate may differ from the exit predicate, as indicated graphically, but in a depth first construction, the entrance and the exit predicate must be the same.

If the conjuncts in a case 1 (paratactic) construction were all depth first, the resulting NAG and surface realization would be as in the following example:

1.4.4 Conjoining depth first structures in a text

(iii) NAG (numbered arcs graph)

(iv) surface realization

For simplicity, the depth first propositions conjoined are minimal, but they may be of any grammatical complexity, as in [1.3.8] or CLaTR 7.6.7. As usual in DBS speak mode graphs, the long lines “−” and “/” are interpreted semantically as conjunct−conjunct and as subject/predicate, respectively, and are reused in the linear notation of the surface realization.

Extrapropositional coordination differs from intrapropositional coordination and functor-argument in general in that it is unidirectional. Apparent forward and backward pairs like John left the house. Then he crossed the street (HBTR 2.5.7) vs. John crossed the street. Before that he left the house (HBTR 2.5.8) are treated as partially related semantic structure ([1.3.7]), and not as a paraphrase ([1.3.2]). Partially related semantic structures are based on the use of inferencing.

Let us turn now to the details of an extrapropositional conjunct with different entrance and exit predicates, corresponding to the middle proposition in case 1 of [1.4.3]

---

12 For a more detailed discussion of extrapropositional coordination see CLaTR 9.4.3, 9.4.4 and NLC Chaps. 11 (hear mode) and 12 (speak mode).
1.4.5 Subject Gapping as a Top Level Conjunct

(iii) NAG (numbered arcs graph)

Having more than one top verb, here buy, peel, and eat, raises the question, which of them should represent the construction in the extrapropositional coordination of a text or dialog (1.4.4) and serve as the entry point for traversing the construction it represents. Because extrapropositional coordination is unidirectional (simplex), it should be the initial verb, here buy (arc 0). However, to reduce the number of traversals and facilitate finding the verb for adding the period, the navigation uses the final verb eat (arc 12) as the exit for a possible continuation to a next proposition.

Let us also consider the other gapping constructions, called predicate gapping (Sect. 5.3) and object gapping (Sect. 5.4). Because predicate gapping has only a single top verb, namely the shared predicate, separate entrance and exit predicates cannot occur. With object gapping, however, there is a similar separation of the entrance and the exit predicate as in 1.4.5.

1.4.6 Object Gapping as a Top Level Conjunct

(iii) NAG (numbered arcs graph)
As in [1.4.5], exiting from *eat* rather than *buy* reduces the number of traversals, and facilitates finding the verb for adding the period and for continuing to the next extrapropositional conjunct.

Finally let us turn to the details of a case 2 (hypotactic) construction of [1.4.3], i.e. the clausal functor-argument embedding of a breadth first construction with different entrance and exit predicates in a depth first proposition. The following example uses the same subject gapping construction as [1.4.5].

1.4.7 SUBJECT GAPPING AS A CLAUSAL SUBJECT

(iii) NAG (numbered arcs graph)

(iv) surface realization

That bob bought an apple peeled a pear and ate a peach amused mary.

The navigation activates the top predicate *amuse* from the top level coordination (not shown). The depth first traversals of arc 1 and 2 reach *bob* as the shared item of a subject gapping construction, which happens to serve as the clausal subject of *amuse*. Triggered by the gap list of *bob* (5.1.17), the arc numbering switches from depth first to breadth first. After traversing the gapped subject clause, the navigation returns via arc 13 from *eat* to *amuse* (i.e. with V/V from the clausal subject back to the predicate) and completes the traversal via arcs 14 and 15 in a depth first manner.

1.5 DBS Laboratory Set-Up for Linguistic Analysis

Mapping a content derived in the hear mode from a language surface back into that same surface in the speak mode is called the DBS laboratory set-up (NLC 3.5.3, CLaTR Sect. 14.4). For example, the operations [1.2.4] and [1.2.5] realize *John* and *bought* from two proplets derived in the hear mode derivation [1.1.3].
The linguistic laboratory set-up (i) requires the hear mode to derive a content which encodes all grammatical information needed for production in the speak mode and (ii) frees the analysis and the software implementation of the speak mode from the aspect of controlling the agent’s behavior towards maintaining a state of balance vis à vis a continuously changing environment (survival). In other words, using content derived by a hear mode derivation for the associated speak mode production allows to temporarily set aside the important control aspects of what to say and how to say it.

At the same time, a software implementing DBS hear and speak grammars in accordance with the laboratory set-up is compatible with an autonomous control in the agent’s cognition: the constraint provided by the laboratory set-up is replaced by guiding the navigation and the inferencing underlying the speak mode (and behavior in general) towards maintaining the agent in a state of balance (6.6.3, 6.6.4; NLC Chap.5; CLaTR Chaps. 5, 6).

Each of the following 24 Sects. 2.1–5.6 analyzes a linguistic sample structure by working off the following to-do list of the DBS laboratory set-up:

1.5.1 To-do list for upscaling within the laboratory set-up

1. Definition of the content for an example surface (e.g. 3.3.1)
2. Graphical hear mode derivation of the content (e.g. 3.3.2)
3. Complete sequence of explicit hear mode operation applications (e.g. 3.3.3–3.3.8)
4. Canonical DBS graph analysis underlying production (e.g. 3.3.9)
5. List of speak mode operation names with associated surface realizations (e.g. 3.3.10)
6. Complete sequence of explicit speak mode operation applications (e.g. 3.3.11–3.3.16)
7. Summary of the system extension and comparison of the hear and speak mode operation applications (e.g. 3.3.17)

Point 1 defines the space in which the linguistic analysis takes place: the surface is the input to the hear mode and the associated content the output, while the content is the input to the speak mode and the original surface the output. Points 2 and 3 specify the details of the hear mode, while points 4–6 specify the details of the speak mode.

Point 7 concludes with a comparison of the hear and the speak mode derivations of the example. It lists the additional operations needed for the construction and compares the number of operation applications needed in the hear and the speak mode.
For instance, example 5.1.1 consists of $n = 14$ word form surfaces (including the period). Therefore, the adverbial hear mode interpretation requires a minimum of $n-1 = 13$ hear mode operations. The associated speak mode derivation, in contrast, requires only 10 operations:

1.5.2 Graph analysis underlying the production of 5.1.1

(iii) NAG (numbered arcs graph)

(iv) surface realization

The reason for the lower number of speak mode operations is the realization of prepositional phrases in a single navigation step.

In contrast, the deterministic hear mode derivation of example 5.5.1 requires 10 operation applications, but the speak mode derivation requires 16:

1.5.3 DBS graph analysis of an unbounded dependency

(iii) NAG (numbered arcs graph)

(iv) surface realization
The reason for the higher number of speak mode operations are the empty traversals for getting to Whom in the lowest object clause and then back empty to the top clause for continuing the navigation."  
Each time the test list has been extended to an additional example and analyzed in the hear and the speak mode in accordance with the to-do list, the declarative specification is tested procedurally by an implementation of the system. Once the complete current test list is debugged such that the hear and the speak mode run without fault, the upscaling cycle is repeated by adding a new construction to the current test list.

1.6 The Test List

This section presents the 24 linguistic examples which will be analyzed in the following chapters. For each example, the grammatical reasons for its construction and its place in the systematic order of presentation are explained.

1.6.1 The Examples of Chap. 2

The general topic of this chapter is the obligatory semantic relations of natural language, namely subject/predicate and object/predicate, at the elementary, phrasal, and clausal levels of grammatical complexity (Sect. 1.3).

1. I slept. Sect. 2.1

This minimal declarative sentence consists of an elementary subject and an elementary predicate. As a most simple complex content, it is used as the initial test list, which consists of a single example.

2. The old dog had been sleeping. Sect. 2.2

Like this example is based on the semantic relation of subject/predicate. It differs in that the subject and the predicate are phrasal rather than elementary. The hear mode analysis shows how the surface, consisting of seven word forms (including the period) is interpreted by adding one analyzed word form after another in a strictly time-linear derivation order, resulting in a content which consists of three proplets. The speak...
mode analysis (2.2.9) shows the time-linear navigation along the semantic relations between these three four proplets and the realization of the seven surfaces.

3. I saw you. Sect. 2.3

This example consists of an elementary subject, predicate, and object. The time-linear hear mode analysis (2.3.2) of the first two word forms is analogous to that of 1. The resulting content consists of three order-free proplets which are connected by the relations subject/predicate and object/predicate. The speak mode analysis (2.3.6) navigates from the predicate to the subject to realize I (pro1), from the subject back to the predicate to realize saw, from the predicate on to the object to realize you, and from the object back to the predicate to realize the period.

4. The big dog saw a small bird. Sect. 2.4

Like 3, this example is based on the semantic relations subject/predicate and object/predicate. It differs in that the subject and the object are phrasal rather than elementary. The time-linear hear mode analysis of the first four word forms is analogous to that of example 2. Because the hear mode (2.4.2) fuses the determiners the and a(n) with their common noun (function word absorption), the seven word form surface results in a five proplet content. In the speak mode (2.4.10), the function words are realized by lexicalization rules which use such proplet values as def(inite) and indef(inite) as input for realizing the different determiners.

5. That Fido barked amused Mary. Sect. 2.5

Like 3 and 4 this example is based on the semantic relations subject/predicate and object/predicate. It differs in that the subject is clausal rather than elementary (1) or phrasal (2). The strictly time-linear hear mode interpretation (2.5.2) assembles the subject clause by adding one word form after another and then connects it to the matrix clause predicate with a V/V relation. The speak mode (2.5.9) navigates from the predicate of the matrix clause to the predicate of the subclause to realize That, and from there to the subclause subject to realize Fido. On the return to the subclause predicate, the speak mode realizes barked, and on the return to the matrix predicate it realizes amused. The time-linear hear and speak mode analysis of the matrix clause remainder is analogous to 4.
6. Mary heard that Fido barked. Sect. 2.6

This example is also based on the subject/predicate and object/predicate relations. It differs from 5 however, in that it is the object rather than the subject which is clausal. Instead of assembling the object clause before connecting it to the main clause predicate, the hear mode derivation (2.6.2) adds it word form by word form in a strictly time-linear fashion. This is enabled by input and output patterns of the hear mode operations which accommodate the local relations within a subclause regardless of whether it precedes of follows the matrix predicate. The speak mode (2.6.9) navigates from hear to mary to realize the subject, back to hear to realize the predicate, from there to the object clause predicate bark to realize that, from there to fido to realize the subclause subject, back to bark to realize the subclause predicate, and finally back to hear to realize the period.

1.6.2 The examples of Chap. 3

The general topic is the optional semantic relations of natural language, namely modifier/modified and conjunct—conjunct, the former at the elementary, phrasal, clausal levels of grammatical complexity, the latter only at the elementary level 15.

7. Perhaps Fido is still sleeping there now. Sect. 3.1

Like 1 this example is based on the subject/predicate relation. In addition, the example shows four elementary adverbial modifiers. In the time-linear hear mode derivation (3.1.2), there is no semantic relation between the first two word forms Perhaps Fido. They are suspended until the finite auxiliary arrives, at which point two operations apply in the same derivation step (suspension compensation): one connects perhaps, the other fido, to the auxiliary. The following adverbial still is also connected to the auxiliary. After the auxiliary and the non-finite main verb have been fused (function word absorption), the adverbials there and now are connected to the predicate. The speak mode (3.1.11) navigates from the predicate to the initial adverbial via V↓A to realize perhaps, back to predicate, on to the subject to realize Fido, back to the predicate to realize is, and so on.

15 For a detailed treatment of clausal coordination see NLC Chaps. 11 (hear mode) and 12 (speak mode).
8. Fido ate the bone on the table. Sect. 3.2

The analysis shows the adverbial (and not the adnominal\footnote{For the analysis of elementary and phrasal adnominal modifiers see NLC Sects. 13.2, 14.2, and 15.3.}) interpretation of the phrasal modifier on the table. After interpreting the initial sentence in analogy to\footnote{Counting the period.} the hear mode (3.2.11) connects the preposition to the predicate, and then adds the determiner and the noun, fusing them into one proplet representing the prepositional phrase. In the speak mode (3.2.11), the values on, def sg, and table of this proplet are used to realize on_the_table from the goal proplet of a single traversal.

9. The dog which saw Mary barked. Sect. 3.3

This example moves from elementary (7) and phrasal (8) adverbial modification to a clausal adnominal modifier, aka a relative clause. By fusing the determiner with dog and the subordinating conjunction with see (function word absorption), the hear mode maps a seven\footnote{Counting the period.} word form surface into a four proplet content (3.3.2). In the content, dog uses see as a modifier, and see uses dog as a modified. In addition, the object\textbackslash{}predicate relation connects mary and see (N\textbackslash{}V), and the subject\textbackslash{}predicate relation connects dog and bark (N\textbackslash{}V). The speak mode (3.3.9) navigates from bark to dog to realize the dog, from dog to see to realize which\_saw, from see to mary to realize Mary, back to see and on to bark to realize barked_.

10. The dog which Mary saw barked. Sect. 3.4

This example differs from\footnote{Counting the period.} in that mary is the subject (and not the object) of the adnominal clause; also dog functions as the implicit object (and not as the implicit subject) of the subclause. This is expressed by the word order which Mary saw (instead of which saw Mary). As a consequence of this word order there is a suspension in the time-linear hear mode derivation (3.4.2): the subordinating conjunction and mary cannot be connected because there is no semantic relation between them. When see arrives, (i) it connects with mary (subject\textbackslash{}predicate) and (ii) fuses with the subordinating conjunction (function word absorption) in the same derivation step (suspension compensation). The speak mode (3.4.10) navigates from bark to dog to realize the dog, from dog to see to realize which, from see to mary to realize Mary, back to see to realize saw, and finally back to bark to realize barked_.
11. When Fido barked Mary laughed.  
This example shows a clausal adverbial. In contradistinction to clausal adnominals (9, 10), the position of clausal adverbials is relatively free (NLC 7.5.1, 7.5.4) and they do not contain any gap. The extrapropositional $V|V$ relation between bark and laugh is coded by the core values laugh in the mdd slot of bark and bark in the mdr slot of laugh. The time-linear hear mode derivation (3.5.2) begins with a suspension because there is no semantic relation between When and Fido. When bark arrives the suspension is compensated by (i) combining fido and bark with subject/predicate and by (ii) fusing the subordinating conjunction and bark (function word absorption). The speak mode derivation (3.5.10) navigates from laugh to bark to realize When, from bark to fido to realize Fido, from fido back to bark to realize barked, from bark back to laugh and on to mary to realize Mary, and from mary back to laugh to realize laughed.

12. Fido, Tucker, and Buster snored loudly.  
This example illustrates the other optional relation in natural language besides modification, namely coordination, represented formally as conjunct−conjunct. The relation is coded by the core value of tucker in the nc (next conjunct) slot of the fido proplet and the core value of fido in the pc (previous conjunct) slot of the tucker proplet, and similarly for the relation between tucker and buster. In the functor-argument structure derived in the hear mode (3.6.2), only the initial conjunct is used for the subject/predicate relation with the predicate (NLC Sects. 8.1, 8.2), while the conjuncts are connected solely by coordination.

1.6.3 THE EXAMPLES OF CHAP. 4

The general topic is the different kinds of arguments in the functor-argument relation, such as prepositional object, discontinuous element, infinitive phrase, and the second argument in copula constructions.

13. Julia put the flowers on the table.  
Modifiers are optional in the sense that an expression without them is still complete, while arguments are obligatory in that an expression without them is incomplete. Prepositional phrases are free to serve as either an optional modifier or an obligatory argument. For example, on the table is an optional modifier in 8 but an obligatory argument here. This is because the verb put takes a subject, a direct object, and a prepositional phrase as
arguments. The hear mode interpretation (4.1.2) is analogous to 8 except that cat attribute of put has the additional valency position mdr' which gets canceled when the preposition is added, making the prepositional phrase an argument. The speak mode derivation (4.1.10) is analogous to 8.

14. The letter made Mary happy.

The verb make is like put in that it may take a modifier as an argument. It differs in that it takes an adnominal, here happy, rather than a prepositional phrase. As in 13, the obligatory nature of the modifier as a valency filler is expressed by happy canceling a mdr' valency position in the cat slot of the verb. The adnominals combining with make are restricted to such mood indicators as {happy, sad, angry, pensive, wary, proud, ...}. For the graphical hear and speak mode derivations see 4.2.2 and 4.2.8 respectively.

15. Fido dug the bone up.

This example raises the question of whether the so-called discontinuous element up is part of the verb (separable prefix) or an argument, similar to happy as in discontinuous make...happy. The DBS analysis takes the second approach. However, while the proplet set of 14 includes a happy proplet (content word), the proplet set of 15 does not include an up proplet (function word); instead, up is absorbed into dig. For the graphical hear and mode derivations see 4.3.2 and 4.3.8 respectively.


The transitive verb try may take a noun, as in try a cookie, or an infinitive, as in try to read a book (CLaTR Sect. 15.4) as its direct object. In the hear mode (4.4.2), the lexical analysis of the intrapropositional conjunction to has the core attribute verb because it absorbs the infinitive, a fnC attribute because the infinitive serves as an argument of the higher predicate, and an arg attribute for the underlying subject of the infinitive, here john. In the speak mode (4.4.9), read is related to try by $\text{V} \setminus \text{V}$ and book is related to read by $\text{N} \setminus \text{V}$.

17. Julia is a doctor.

The core value of the auxiliary be is a substitution variable, here $v_1$. When used as a function word in a complex verb construction, the substitution variable is replaced by the core value of the infinite main verb (function word absorption). In the present example, however, the auxiliary
is used as a copula taking a noun as its second argument. Citing conflicting morphological evidence of English, the question of whether the second noun should be analyzed as a nominative or as oblique is answered in favor of the latter. In this way, the hear mode derivation \((4.5.2)\) may be based on the standard operations SBJ×PRD and PRD×OBJ. The speak mode analysis \((4.5.7)\) is also based on the standard analysis of a transitive sentence.

18. Fido is hungry. 

The cat values of the auxiliary form is include two valency positions, ns3’ and be’. The first is canceled by the subject, the second by either a noun interpreted as an object, as in 17, or by an adnominal. In the present example, (i) the valency position be’ is canceled by the adnominal hungry and (ii) the core values of _v_I and hungry are cross-copied into the respective mdd and mdr slots. This simultaneous valency canceling and cross-copying characterizes hungry as an argument (and not as an optional modifier), analogous to 14. For the graphical hear and speak mode derivations see 4.6.2 and 4.6.6 respectively.

1.6.4 The examples of Chap. 5.

The general topic is grammatical structures which are commonly analyzed as a form of recursion, i.e. as substitution in a self-similar way, e.g. \(x = x+1\), \(NP \rightarrow NP \ S\), or \(N \rightarrow A \ N\). Here, in contrast, the structures are analyzed as repeated continuations based on address values, and not as recursive substitution. The examples are prepositional clause repetition, subject gapping, predicate gapping, object gapping, unbounded object clause repetition (aka unbounded dependency), and adnominal subclause repetition.

19. Fido ate the bone on the table under the tree in the garden...

The beginning of this example is the same as 8. The difference is the continuation with repeated prepositional phrases. Used as adnominal modifiers, they are attached by cross-copying between the mdr slot of the modified and the mdd slot of the modifier, but without canceling a valency position (unlike 13, 14, and 18). For the graphical hear and speak mode derivations see 5.1.2 and 5.1.12 respectively.
20. Bob bought an apple, $\emptyset$ peeled a pear, ..., and $\emptyset$ ate a peach.  
   Sect. 5.2

In this example of subject gapping, the hear mode (5.2.2) constructs the gap list \textit{buy, peel, eat} incrementally in the shared subject \textit{bob}. It is a repeating construction because there is no grammatical limit on the number of gapped items. The speak mode (5.2.15) uses the shared subject repeatedly as a valency filler, but with only a single (initial) surface realization.

21. Bob bought an apple, Jim $\emptyset$ a pear, ..., and Bill $\emptyset$ a peach.  
   Sect. 5.3

In this example of predicate gapping, the gap list \textit{bob apple, jim pear, bill peach} is constructed incrementally in the shared predicate \textit{buy} by the hear mode (5.3.2). Like all gapping constructions, it is intrapositional. The speak mode (5.2.15) uses the shared predicate repeatedly as the valency carrier, but with only a single (initial) surface realization.

22. Bob bought $\emptyset$, Jim peeled $\emptyset$, ..., and Bill ate a peach.  
   Sect. 5.4

While the shared item in subject and predicate gapping precedes the gaps, the shared item follows the gaps in object gapping, as shown by the shared object \textit{peach} of this example. For building up the gap list \textit{buy, peel, eat}, the hear mode (5.4.2) uses a storage variable and moves its content into the shared object \textit{peach} as soon as it becomes available. As in subject gapping (20), the speak mode (5.4.13) ends the production with the last rather than the first predicate.

23. Whom does John say that Bill believes ... that Mary loves?  
   Sect. 5.5

This construction has been called an “unbounded dependency” between the initial \textit{Whom} and its grammatical role as the object of the final object sentence. In DBS, this construction is analyzed as an unbounded suspension. The time-linear hear mode (5.5.2) derivation may continue to add object clauses as long as the verb of last object clause is what is called a \textit{mental state} verb in lexicography. The speak mode (5.5.17) uses two empty traversals (3, 6) to access \textit{whom} as the object of the last (lowest) object sentence and two (10, 11) to return to the top for continuing the derivation.

24. Mary saw the man who loves the woman ... who fed Fido.  
   Sect. 5.6

In this example of repeating adnominal subclauses (aka relative clauses) with subject gaps, the object noun \textit{man} is in a \textit{mdr|mdd} relation with
love (representing the first subclause) and the object noun woman is in a mdr|mdd relation with feed (representing the second subclause). In the hear mode (5.6.2), the N|V relations between the head noun and the modifier clause are established by cross-copying and based on addresses. The repetition of adnominal subclauses may be continued indefinitely as long as the agents’ cognitive reserves permit. The speak mode (5.6.14) returns with five empty traversals (8–12) to the top predicate see to realize the period.

The operations defined for the explicit hear and speak mode derivations of each of the 24 examples are summarized as DBS.Hear (Sect. 6.3) and DBS.Speak (Sect. 6.5). The operations are presented as declarative specifications (CLaTR Sect. 1.5), i.e. at a level of formal detail directly above the software definitions in a programming language, but without the accidental properties of a procedural implementation.

The linguistic analysis is embedded into the functional flow of agent-based DBS. As a specialization of recognition, the hear mode maps the input of external language surfaces into content. The hear mode operations apply at the now front, where they replace their input with their output. The now front is cleared in regular intervals by moving it into free memory space, leaving the content derived behind as permanent sediment (loom-like clearance).

The operations of the think mode consist of (i) selective activation and (ii) inferencing. Selective activation navigates along semantic relations coded by address between items stored in memory. Inferencing refers to old content by address, and derives and stores new content at the now front. By using addresses, neither selective activation nor inferencing touch the stored content. Within the DBS laboratory set-up (Sect. 1.5), which is used here throughout, think mode operations are confined to the selective activation of content derived in the hear mode.

In the speak mode, think mode operations are supplemented with lexicalization rules which are placed in the sur slot of the goal pattern. If a continuation proplet is activated by matching the goal pattern of a think mode operation, the lexicalization rule uses its core, cat, and sem values for language-dependent automatic word form production.
Use of Variables and their Markings

Lower case Greek letters are used as variables for single core or continuation values. For example, \( \text{noun: } \alpha \) may be used to match the core value \( \text{noun: dog} \) (2.2.4), and \( \text{mdr: } \alpha \) to match the continuation value \( \text{mdr: old} \) (2.2.3). A variable like \( \alpha \) requires a counterpart at the content level. This is in contradistinction to uppercase roman letters like X, Y, Z, which are used as variables for zero, one, or more unspecified constant values or substitution variables.

Variables and their constant counterparts may be marked in three ways.

1.6.5 The \#X-, X-, and .X- Markings of Variables

1. \#X
   The \#-marking of variables is used differently in the hear mode and the speak mode:
   In the hear mode, \#-marking is used to cancel valency positions. For example, \( \text{cat: } \#n' \ a' \ v \) indicates that the nominative valency position has been filled, while the accusative position is still open (e.g. 2.3.4).
   In the speak mode, \#-marking is applied to continuation values and indicates that they have been traversed at least once. The pattern matching is asymmetric: if a variable is \#-marked, the corresponding value at the content level must be \#-marked as well (5.5.11 5.5.18); a variable which is not \#-marked, in contrast, may be matched by a \#-marked (5.5.31) or an unmarked (5.5.25) value at the content level.

2. X
   A variable with a solid underline requires at least one matching value at the content level in the hear (2.3.4 2.4.6 2.5.7 ..., 5.6.8 5.6.12) and the speak (5.2.21 5.2.25) mode.

3. .X
   A variable with a dotted underline matches values with (2.6.16 5.5.33 5.5.34) and without (5.5.23 5.5.24) an \#-marking in the speak mode.

4. &K
   The pattern matching with a \&-marked prn variable is asymmetric like \#-marking in the speak mode (1).

The purpose of 1 is to control the time-linear flow of derivations and to prevent unwanted repetitions, 2 serves to control the application of a couple of operations in gapping constructions, 3 and 4 each save an operation.
2. Obligatory Functor-Arguments

There are two basic semantic relations of structure[1] in natural language, (i) functor-argument and (ii) coordination[2] at the elementary, phrasal, and clausal levels of grammatical complexity (1.3.7). Functor-argument divides into the obligatory relations of subject/predicate and object/predicate, and the optional relations of modifier|modified, which in turn divide into adnominal|noun and adverbial|verb. The obligatory relations are determined by the valency structure of the verb[3]. This chapter presents the DBS treatment of six obligatory functor-argument constructions in the hear and the speak mode.

2.1 DBS.1: Elementary Subject/Predicate

As a maximally simple subject/predicate construction consider the example I slept. and its content:

2.1.1 CONTENT OF THE SURFACE I slept.

The proplets are held together by a common prn (proposition number) value, here 1. The subject/predicate relation is coded by (i) the value pro1 of the core attribute noun in the first proplet, which reappears as the value of the

---

1 The semantic relations of grammatical structure differ from the semantic relations of meaning such as hypernym, hyponym, synonym, antonym, etc., and of content such as cause_and_effect.
2 In symbolic logic, extrapropositional coordination is formalized as propositional calculus, while functor-argument is grafted on top of propositional calculus in predicate calculus.
3 In linguistics, an argument is called obligatory if a sentence is incomplete without it; for example John crossed. is incomplete because cross is missing its obligatory grammatical object.
continuation attribute \textit{arg} of the second proplet, and (ii) the value \textit{sleep} of the second proplet’s core attribute \textit{verb}, which reappears as the value of the continuation attribute \textit{fnc} of the first proplet (resulting from cross-copying).

The input to the hear mode is a time-linear sequence of unanalyzed external word form surfaces, i.e., sound waves in spoken and dots on paper in written language (raw input). Automatic word form recognition (CLaTR Sect. 2.2, NLC 11.1) matches each unanalyzed surface with the \textit{sur} value of a lexical proplet provided by the agent’s memory.

Once a word form has been recognized in this way, the surface plays no role in subsequent processing and may be omitted. This is because DBS processing is run via the core and continuation values, making the surfaces superfluous and the contents relatively language-independent. Also, if the surfaces were still present when the agent switches into the speak mode, a language-dependent automatic word form production as the counterpart to automatic word form recognition would be of little purpose.

The operations of the hear mode either concatenate or fuse the lexical proplets provided by automatic word form recognition. For example, if the surface \textit{slept} would be of little purpose.

The operations of the hear mode either concatenate or fuse the lexical proplets into a content by means of cross-copying:

\begin{align*}
\text{SBJ} \times \text{PRD} (\text{line 1 in 2.1.4})
\end{align*}

The operation consists of two pattern proplets for matching an input, the connective \textit{⇒}, and two pattern proplets for deriving an output. Underneath the
pattern level, above the matching frontier, are the agreement conditions, consisting of (i) variable restrictions and (ii) conditions. The agreement condition in 2.1.2 ensures, for example, that I (s1) walk (n-s3 v) is accepted as grammatical, but I (s1) walks (ns3 v) is rejected as ungrammatical.

An output is derived by binding constants of the content level to corresponding variables of the operations level in the input. For example, in 2.1.2 the constants pro1 and sleep are bound to the variables α and β in the input, which results in adding the values sleep and pro1 to the fnc and arg slots of the output, thus performing cross-copying computationally. Defining the operations explicitly serves as the declarative specification of the DBS software.

To complete the content in 2.1.1, the hear mode derivation must add the period with the operation S∪IP:

2.1.3 ABSORBING • INTO sleep WITH S∪IP (line 2 in 2.1.4)

When analyzing an unambiguous sentence with n word forms (including interpunctuation), the number of hear mode operations like 2.1.2 and 2.1.3 will be exactly n-1. Even for short input, it is helpful if the explicit operation sequence (to-do list 1.5.1, 3) is preceded by a concise graphical summary in the canonical graph format of the DBS hear mode (to-do list 1.5.1, 2), such as the following:

---

4 This does not preclude remembering the wording used by a partner in discourse. It just means that the cycle of communication modeled in DBS maps a time-linear sequence of unanalyzed surfaces into a content in the hear mode and a content into a sequence of unanalyzed surfaces in the speak mode.

5 The letter h preceding the number in (h1) refers to an operation in the DBS hear grammar in Sect. 6.3. In contrast, the letter s preceding the number (s1) in 2.1.6, for example, refers to an operation of the speak mode grammar in Sect. 6.5.

6 For the valency positions of English main verbs and auxiliaries see NLC, A.5.1–A.5.6. For the categories of names, pronouns, and determiners as valency fillers see NLC, A.4.1–A.4.3.

7 In some operations, the agreement conditions consist only of variable restrictions, without conditions.
2.1.4 Graphical hear mode derivation of the content 2.1.1

unanalyzed surface

1 slept

automatic word form recognition

<table>
<thead>
<tr>
<th>sur:</th>
<th>noun: pro1</th>
<th>verb: sleep</th>
<th>sur:</th>
</tr>
</thead>
<tbody>
<tr>
<td>prn:</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>cat:</td>
<td>s1</td>
<td>n’ v</td>
<td></td>
</tr>
<tr>
<td>sem:</td>
<td>sg</td>
<td>past</td>
<td></td>
</tr>
</tbody>
</table>

syntactic-semantic parsing

| sur: | noun: pro1 | verb: sleep | cross-copying
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>prn:</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>cat:</td>
<td>s1</td>
<td>n’ v</td>
<td></td>
</tr>
<tr>
<td>sem:</td>
<td>sg</td>
<td>past</td>
<td></td>
</tr>
<tr>
<td>fnc:</td>
<td>sleep</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

result

| sur: | noun: pro1 | verb: sleep | absorption
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>prn:</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>cat:</td>
<td>s1</td>
<td>n’ v</td>
<td>v_1</td>
</tr>
<tr>
<td>sem:</td>
<td>sg</td>
<td>past</td>
<td></td>
</tr>
<tr>
<td>fnc:</td>
<td>sleep</td>
<td></td>
<td>pro1</td>
</tr>
</tbody>
</table>

Compared to 2.1.1 the empty attributes mdr, nc and pc have been omitted for reasons of space and transparency. The derivation is time-linear, as shown by the addition of exactly one next word in each derivation step (here only two). It is also surface compositional because (i) each lexical proplet has a sur value and (ii) there are no surfaces without an explicit lexical proplet.

The cross-copying performed by 2.1.2 is indicated graphically in line 1 by diagonal arrows. Line 2 shows (i) the result of the operation and (ii) the addition of a next word, here the period, with 2.1.3. The period proplet is absorbed into the sleep proplet, as shown by the horizontal arrow. The input of a hear mode operation is replaced by its output, as shown in the next derivation step.

Next we turn to the speak mode. In accordance with the laboratory set-up of DBS (Sect. 1.5), the speak mode maps the content derived in the hear mode automatically back into the original surface. For characterizing a speak mode derivation graphically, DBS uses a second canonical format, illustrated below with the navigation through the content derived in 2.1.4.
2.1.5 Graph analysis underlying production of 2.1.1

(i) SRG (semantic relations graph)  
(ii) signature  
(iii) NAG (numbered arcs graph)  
(iv) surface realization

The (i) SRG, (ii) signature, (iii) NAG, and (iv) surface realization provide four different views on the content. The SRG uses the core values of the content 2.1.1 while the signature uses the core attributes. The NAG adds numbered arcs to the SRG. The top row of the surface realization shows the arc number in which the surface directly underneath is realized from the goal proplet; the bottom row characterizes the operation which traversed the semantic relation. The lines of the graphs (i)–(iii) correspond to the slashes in the linear notation of (iv).

While the semantic relations in the SRG and the signature are static, they are dynamic in the NAG and the surface realization. The dynamic aspect is in the traversal of a relation from one node to the next. Because a relation may be traversed in two directions, there are two arcs for the subject/predicate relation in 2.1.5 and the two operations V/N and N/V for their traversal (1.2.2)8

As the first step of activating the content 2.1.1, the operation V/N navigates from the V via the “/” relation to N:

2.1.6 Navigating with V/N from sleep to pro1 (arc 1)

8 The arrow heads are omitted in the bottom row of the surface realization for simplicity and because the direction may be inferred from the associated arc number in the top row.
Based on the $N/V$ relation established in the hear mode between the `sleep` and the `pro1` proplets by $\text{SBJ} \times \text{PRD}$ (2.1.2), the navigation uses the $V/N$ speak mode operation (2.1.6) to travel from $V$ to $N$. The word form production rule lexnoun in the sur slot of the goal pattern (pattern level) uses the core and the cat values of the goal proplet (content level) to realize the surface `I` in the sur slot of the `pro1` proplet.

The other operation needed for activating the content (2.1.1) is $N/V$. It also navigates along the semantic relation $N/V$ established by $\text{SBJ} \times \text{PRD}$ in (2.1.2) but in the opposite direction compared to (2.1.6).

### 2.1.7 Navigating with $N/V$ from `pro1` back to `sleep` (arc 2)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: $\beta$</td>
<td>sur: slept_..</td>
<td><code>-mark $\alpha$ in the fnc slot of proplet $\beta$</code></td>
</tr>
<tr>
<td>fnc: $\alpha$</td>
<td>verb: sleep</td>
<td><code>decl</code></td>
</tr>
<tr>
<td>mdr: Z</td>
<td>cat: <code>#n</code> <code>decl</code></td>
<td><code>past</code></td>
</tr>
<tr>
<td>prn: K</td>
<td>arg: <code>#pro1</code></td>
<td><code>#</code></td>
</tr>
<tr>
<td>Z is NIL or elementary and <code>-marked</code></td>
<td>mdr:</td>
<td><code>nc:</code></td>
</tr>
<tr>
<td></td>
<td>pc:</td>
<td><code>prn: 1</code></td>
</tr>
<tr>
<td></td>
<td>prn: 1</td>
<td><code>-mark $\alpha$ in the fnc slot of proplet $\beta$</code></td>
</tr>
</tbody>
</table>

Here the lexicalization rule is lexverb. Using the values `sleep`, `decl`, and `past` of the goal proplet, it produces the surface `slept_..`.

There are navigation operations which produce one surface, e.g. (2.1.6) two surfaces, e.g. (2.1.7) three or more surfaces, e.g. (2.2.14) or no surface at all, e.g. (3.1.14). It is therefore interesting linguistically to see at one glance which surfaces are produced in which traversal (to-do list (1.5.1), especially in the speak mode production of longer sentences. As a maximally simple example consider the list of operations underlying the production of the English surface in (2.1.1).

### 2.1.8 Sequence of operation names and surface realizations

<table>
<thead>
<tr>
<th>arc 1: $V/N$ from <code>sleep</code> to <code>pro1</code></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>arc 2: $N/V$ from <code>pro1</code> to <code>sleep</code></td>
<td><code>slept_..</code></td>
<td>2.1.6</td>
<td>2.1.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*In the static graph representation of a semantic relation, the lower node in the graph precedes in the corresponding linear notation. In the dynamic NAG representation showing the traversal of a semantic relation, in contrast, it is the start node which precedes in the corresponding linear notation.*
Column 1 specifies the arc of the traversal in the associated NAG, 2 names the operation, 3 describes the traversal, 4 specifies the surface(s) produced, and 5 states the reference number of the explicit operation application.

Let us follow standard procedure and conclude this section with point 7 on the to-do list. The maximally simple DBS.1 grammar for the class of elementary subjects and elementary predicates required the definition of the two hear mode operations $\text{SBJ} \times \text{PRD} (h1)$ and $\text{S} \cup \text{IP} (h13)$, and the two speak mode operations $\text{V} / \text{N} (s1)$ and $\text{N} / \text{V} (s2)$. They happen to be equal in number, but the incoming and the outgoing surfaces are segmented differently:

2.1.9 Comparing Surface Segmentation in Hear and Speak Mode

**Hear mode:**

\[
\begin{array}{cc}
1 & 2 \\
I & \times \text{slept} \cup .
\end{array}
\]

**Speak mode:**

\[
\begin{array}{cc}
1 & 2 \\
\text{V/N} & \text{N/V} \text{slept}_.
\end{array}
\]

In the hear mode, the numerals above the operations correspond to the line numbers in the graphical derivation. In the speak mode, the numerals refer to the corresponding arc number of the NAG. While automatic word form recognition always takes a single surface as input, the number of word forms produced by automatic word form production will vary. Therefore the interpretation of an example may require a different number of operations in the hear mode than the corresponding production in the speak mode.

### 2.2 DBS.2: Phrasal Subject and Phrasal Predicate

The previous section presented the N/V relation with an elementary argument and functor. The present section extends this analysis to a phrasal argument and functor. Following traditional linguistic method, we replace the elementary subject I with phrasal the old dog and the elementary predicate slept with phrasal had been sleeping. We begin with the content of the example:

2.2.1 Content of the Surface

The old dog had been sleeping.

![Surface representation](image)
Because the surface has three content words, i.e. dog, old, and sleep, the content consists of three proplets. The function words The, has, been, and . are absorbed into the proplets dog and sleep in such a manner that they may be easily resurrected by automatic word form production in the speak mode.

Compared to 2.1.1 the content illustrates the additional modifier-modified relation between the adnominal old and the noun dog (A\N). The relation is coded by cross-copying the core value of old into the mdr slot of dog and the core value of dog into the mdd slot of old.

The canonical graph derivation of the example shows cross-copying in lines 1 and 3, and function word absorption in the lines 2, 4, 5, and 6:

2.2.2 Graphical hear mode derivation of the content 2.2.1
The first hear mode operation applying in the above derivation is called DET×ADN. Based on the core features \([\text{noun: } N_n] \) and \([\text{adj: } \alpha] \) of the input patterns, the operation matches a determiner and an adnominal. At the content level, the substitution variable \(n_1\) stems from the lexical proplet of the determiner. After cross-copying, \(n_1\) occurs also in the mdd slot of \(\text{old}\) and \(\text{old}\) occurs in the mdr slot of \(n_1\).

The application of DET×ADN in line 1 may be shown as follows:

### 2.2.3 CROSS-COPYING \(n_1\) AND \(\text{old}\) WITH DET×ADN (line 1)

\[
\begin{array}{c}
\text{DET} \times \text{ADN} \\
\text{pattern level} \\
\text{[noun: } N_n] \quad \text{adj: } \alpha \\
\quad \text{cat: } \text{CN'} \text{ NP} \\
\quad \text{mdr: } \text{K} \\
\quad \text{prn: } K \\
\quad \uparrow \\
\text{content level} \\
\quad \text{sur: } \text{The noun: } n_1 \\
\quad \text{cat: nm'} \text{ NP} \\
\quad \text{sem: def} \\
\quad \text{fnc: } \\
\quad \text{mdr: } \\
\quad \text{nc: } \\
\quad \text{pc: } \\
\quad \text{prn: } 2 \\
\end{array}
\[
\begin{array}{c}
\text{DET} \times \text{ADN} \\
\text{[noun: } N_n] \quad \text{adj: } \alpha \\
\quad \text{cat: } \text{ADN} \\
\quad \text{mdr: } N_n \\
\quad \text{nc: } \text{ea+} \\
\quad \text{prn: } K \\
\downarrow \\
\text{result} \\
\quad \text{sur: } \text{sleeping verb: } v_2 \\
\quad \text{cat: } \#n' \#hv' \#be' v \\
\quad \text{fnc: } \\
\quad \text{mdr: } \\
\quad \text{nc: } \\
\quad \text{pc: } \\
\quad \text{prn: } 2 \\
\end{array}
\]
The proplet representing the is called \( n_1 \) in the heading of 2.2.3 because the hear, think, and speak mode operations of DBS refer to proplets by their core value (and not by their surface value, with names as a partial exception).

The next operation to apply is \( \text{DET} \cup \text{CN} \). It shows a function word absorption together with a simultaneous substitution. The substitution applies automatically to all occurrences of a variable, here \( n_1 \), at the now front:

### 2.2.4 Absorbing dog into \( n_1 \) with \( \text{DET} \cup \text{CN} \) (line 2)

The old proplet shows the effect of the simultaneous substitution.

The next word proplet \( v_1 \) (for had) activates the operation \( \text{SBJ} \times \text{PRD} \) by matching its second input pattern and finds dog at the now front for its first. The operation disregards the adnominal old (cf. arc 4 in 2.2.9).

### 2.2.5 Cross-copyping dog and had with \( \text{SBJ} \times \text{PRD} \) (line 3)

Agreement conditions as in 2.11.
The SBJ × PRD applications in 2.1.2 and here differ in that here the core value of the verb proplet is the substitution variable v_1, and not a constant.

The following operation 2.2.6 combines the finite auxiliary had and the non-finite form been. It must accommodate the agreement conditions between the three English auxiliaries be, have, and do and the non-finite verb forms they combine with, as well as their agreement conditions regarding person and number, e.g. am, are, is, was, were, etc., between the finite auxiliary and its nominative argument. Consider the following variants:

John is saying...
John was saying...
John will say...
John has said...
John had said...
John does say...
... I am saying...
... you are saying...
... he is saying...

Capturing these variations in the form of variable restrictions and conditions is one of those linguistic “watch maker” tasks typical of DBS.

Next, the hear mode must add been to The old dog has:

2.2.6 ABSORBING been INTO v_1 WITH AUX∪NFV (line 4)

The core value of the next word form been is an automatically incremented V_n value. For the operation to apply, the AUX’ value (valency position) in the first proplet and the AUX value (valency filler) in the second must agree. The
two input proplets are fused into one in the output (function word absorption). For production in the speak mode, their respective \textit{cat} and \textit{sem} values are preserved in the output proplet.

The next word provided by automatic word form recognition, i.e. the non-finite main verb \textit{sle}eping, triggers another application of \textit{AUX} $\cup$ \textit{NFV}:

\begin{center}
\textbf{2.2.7 ABSORBING } \textit{sle}eping \textit{INTO } \mathit{v}_2 \textit{WITH } \textit{AUX} $\cup$ \textit{NFV} \textit{(line 5)}
\end{center}

\begin{center}
\begin{tabular}{c|c|c}
\hline
pattern level & content level & pattern level \hline
\textit{AUX} $\cup$ \textit{NFV} \textit{(h17)} & \textit{AUX} $\cup$ \textit{NFV} (h18) & \textit{AUX} $\cup$ \textit{NFV} (h19) \\
\hline
verb: V\text{\_}n & verb: \alpha & verb: \beta \\
\textit{cat: #W'} \textit{AUX'} \textit{VT} & \textit{cat: #W'} \textit{#AUX'} \textit{Y VT} & \textit{cat: #X'} \textit{VT} \\
\textit{sem: X} & \textit{sem: Z} & \textit{prn: K} \\
prn: K & & \\
\hline
\end{tabular}
\end{center}

Here the \textit{cat} value \textit{be} of the \textit{slee}p proplet cancels the \textit{be}' valency position of \textit{has\_been}. Again the two input proplets are fused into one in the output.

The hear mode derivation concludes with an application of \textit{S} $\cup$ \textit{IP}:

\begin{center}
\textbf{2.2.8 ABSORBING \textit{.} INTO } \textit{slee}p \textit{WITH } \textit{S} $\cup$ \textit{IP} \textit{(line 6)}
\end{center}

By agreement conditions omitted in accordance with the DBS laboratory set-up, we continue with the associated speak mode analysis, beginning with the canonical DBS graph analysis (point 4 on the to-do list 1.5.1):
2.2.9 Graph analysis underlying production of

(i) SRG (semantic relations graph)

(ii) signature

(iii) NAG (numbered arcs graph)

(iv) surface realization

Because *old* is a content word, it is connected graphically to *dog* by the A|N relation. The verb forms *has* and *been*, in contrast, are function words and were fused with *sleeping* into a single node by the hear mode derivation in 2.2.5–2.2.7.

Following standard procedure, we continue with point 5 on the to-do list 1.5.1, i.e. the list of speak mode operation names with surface realizations:

2.2.10 Sequence of operation names and surface realizations

arc 1: V/N from *sleep* to *dog*  
arc 2: N\downarrow A from *dog* to *old*  
arc 3: A\uparrow N from *old* to *dog*  
arc 4: N/V from *dog* to *sleep* *has_been_sleeping_*.  

The surface *The old dog* is a discontinuous structure, consisting of discontinuous *The ... dog* and intervening *old*. In the speak mode, this phenomenon is treated by utilizing the two directions of traversing an arc. For example, *the* is realized by the operation V/N (downward arc 1) and *dog* by A\uparrow N (upward arc 3). The intervening element is realized by the N\downarrow A (arc 2) traversal of the A|N relation. As shown by the surface realization of 2.2.9 and the fourth column of 2.2.10, each of the four arcs happen to produce a surface.

For a more detailed grammatical analysis let us turn now to the sequence of explicit speak mode operation applications. Leaving verb-less constructions
such as WH-answers, headlines, exlaimations, etc. aside, a speak mode derivation begins with the top predicate.\footnote{If there is no unique top predicate, as in subject (1.4.5) and object (1.4.6) gapping, the navigation enters the content with the first of several predicates of equal rank and exits with the last.}

2.2.11 \textbf{Navigating with $V/N$ from sleep to dog (arc 1)}

\[\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \text{#-mark } \beta \text{ in the arg slot of proplet } \alpha \\
\hline
\text{sur: sleep noun: } \beta \text{ fnc: sleep prn: } K & \text{sur: The noun: dog cat: snp sem: def sg fnc: fnc: prn: } 2 \\
\text{verb: sleep arg: } \beta X \text{ prn: } K & \text{sur: The noun: dog cat: snp sem: def sg fnc: fnc: prn: } 2 \\
\end{array}\]

The clause to the right of the operation is called an instruction \[6.4.1\].

As shown by the value \textit{old} in the \texttt{mdr} slot of the goal proplet, the noun \textit{dog} is modified by an elementary adnominal. In such a case, the lexicalization rule \texttt{lexnoun} of $V/N$ realizes only the determiner, based on the \texttt{cat} and \texttt{sem} values of the goal proplet. The realization of the common noun surface \textit{dog} is left to a return navigation, here $A \uparrow N$ \[2.2.13\].

The next operation $N \downarrow A$ moves from an N to an A via the modifier|modified relation. The application is controlled by the core attribute \texttt{noun} of the first and \texttt{adj} of the second input pattern:

2.2.12 \textbf{Navigating with $N \downarrow A$ from dog to old (arc 2)}

\[\begin{array}{c|c}
\text{pattern level} & \text{content level} \\
\hline
\text{sur: lexadj(\hat{\gamma})} & \text{sur: lexadj(\hat{\gamma})} \\
\text{adj: } \hat{\gamma} \text{ mdd: } \beta \text{ prn: } K & \text{adj: } \hat{\gamma} \text{ mdd: } \beta \text{ prn: } K \\
\end{array}\]
Finding the goal proplet *old* in the agent’s memory is based on the continuation value *old* in the *mdr* slot of *dog*, confirmed by the *dog* value in the *mdd* slot of *old*, and on the *prn* value, here 2, common to the two input proplets.

The navigation returns from the adnominal modifier to the noun with **A↑N**:

### 2.2.13 Navigating with **A↑N** from *old* back to *dog* (arc 3)

```
A↑N (\(\downarrow\))

pattern

level

\[\begin{array}{c}
\text{adj: } \gamma \\
\text{mdd: } \beta \\
\text{nc: } Z \\
\text{prn: } K
\end{array}\]

\[\Rightarrow\]

\[\begin{array}{c}
\text{sur: lexnoun(\(\hat{\beta}\))} \\
\text{noun: } \beta \\
\text{mdr: } \#\gamma \\
\text{prn: } K
\end{array}\]

content

level

\[\begin{array}{c}
\text{sur: } \text{dog} \\
\text{noun: } \text{dog} \\
\text{cat: snp} \\
\text{sem: def sg} \\
\text{fnc: sleep} \\
\text{mdr: } \#\text{old} \\
\text{nc: } \text{pc: } \text{prn: } 2
\end{array}\]

\[\uparrow\]

\[\downarrow\]

The condition (6.4.1) deals with a possible coordination of elementary adnominals ([nc: Z]), as in *old, black, sleek, ...*. The return from an elementary adnominal to the modified noun may occur only if (i) there is no coordination or (ii) all conjuncts have been #-marked. In 2.2.13, the first condition is fulfilled: the nc slot of *old* is empty.

The last navigation step activating the content 2.1.1 and realizing the surface *has_be^n_sleeping_* is performed by the speak mode operation **N/V**:

### 2.2.14 Navigating with **N/V** from *dog* back to *sleep* (arc 4)

```
N/V (\(\downarrow\))

pattern

level

\[\begin{array}{c}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K
\end{array}\]

\[\Rightarrow\]

\[\begin{array}{c}
\text{sur: lexverb(\(\hat{\alpha}\))} \\
\text{verb: } \alpha \\
\text{arg: } \#\beta X \\
\text{prn: } K
\end{array}\]

content

level

\[\begin{array}{c}
\text{sur: } \text{has_be^n_sleeping_*} \\
\text{verb: sleep} \\
\text{cat: } \#n' \#iv' \#be' decl \\
\text{sem: past perf prog} \\
\text{arg: } \#\text{dog} \\
\text{mdr: } \text{nc: } \text{pc: } \text{prn: } 2
\end{array}\]

\[\uparrow\]

\[\downarrow\]

Z is NIL, or elementary and #-marked.

#-mark \(\alpha\) in the fnc slot of proplet \(\beta\)
The condition ensures that a return from the subject to the predicate via N/V is only permitted if the noun either doesn’t have an (elementary, phrasal, or clausal) modifier or the mdr value of the noun is #-marked.

Extending DBS.1 to an example class with phrasal subjects and predicates required the definition of three hear mode operations, namely DET × ADN (h41), DET ∪ CN (h37), and AUX ∪ NFV (h15), and two speak mode operations, namely N↓A (h41), and A↑N (h12). The hear mode derivation used six operation applications, while the speak mode derivation used four:

2.2.15 Comparing Hear and Speak Mode Surface Segmentation

hear mode: 1 2 3 4 5 6
   The × old U dog × had U been U sleeping U .

speak mode: 1 2 3 4
   V/N the N↓A old A↑N dog N/V has Been_sleeping .

Here the number of speak mode operations is lower than the number of the hear mode operations. We note in passing that automatic word form recognition (hear mode) and production (speak mode), presented here only in outline, requires substantial empirical work, but does not jeopardize the low computational complexity of the overall DBS software system.

2.3 DBS.3: Elementary Subject/Predicate/Object

The other obligatory functor-argument relation in natural language besides subject/predicate is object/predicate. If there are two objects in English, one is the indirect and the other the direct object. In accordance with point 1 on the to-do list [5.1] we begin by presenting the content for the example surface:

2.3.1 Content of the Surface I saw you.

The proplets are order-free, but for showing them some order must be used. Here, the first proplet represents the subject, the second the predicate, and the third the object. The proplets are held together by a common prn value, here 3.
As a result of cross-copying, the core value pro1 of the subject reappears as the first value in the arg attribute of the predicate and the core value of the predicate reappears in the fnc slot of the subject. The core value pro2 of the object reappears as the second value in the arg attribute of the predicate and the core value of the predicate reappears in the fnc slot of the object.

The content 2.3.1 results from the following DBS hear mode derivation:

2.3.2 GRAPHICAL HEAR MODE DERIVATION OF THE CONTENT 2.3.1

The cross-copyings are shown in lines 1 and 2.

We follow standard procedure by continuing with point 3 on the to-do list 1.5.1, i.e. the sequence of explicit hear mode operation applications which underlie the time-linear hear mode derivation 2.3.2. The first operation applying in the above derivation (line 1) is standard SBJ×PRD:
2.3.3 Cross-copying pro1 and saw with SBJ×PRD (Line 1)

SBJ×PRD

### pattern level

- **cat:** NP
- **cat:** NP
- **arg:** K
- **arg:** K
- **fnc:** see
- **fnc:** see
- **verb:** β
- **verb:** β
- **noun:** α
- **noun:** α

### content level

<table>
<thead>
<tr>
<th>sur:</th>
<th>saw</th>
<th>sur:</th>
<th>saw</th>
<th>sur:</th>
<th>saw</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat:</td>
<td>pro1</td>
<td>cat:</td>
<td>pro1</td>
<td>cat:</td>
<td>pro1</td>
</tr>
<tr>
<td>noun:</td>
<td>see</td>
<td>noun:</td>
<td>see</td>
<td>noun:</td>
<td>see</td>
</tr>
<tr>
<td>verb:</td>
<td>see</td>
<td>verb:</td>
<td>see</td>
<td>verb:</td>
<td>see</td>
</tr>
<tr>
<td>prn:</td>
<td>K</td>
<td>prn:</td>
<td>K</td>
<td>prn:</td>
<td>K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sur:</th>
<th>you</th>
<th>sur:</th>
<th>you</th>
<th>sur:</th>
<th>you</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat:</td>
<td>pro2</td>
<td>cat:</td>
<td>pro2</td>
<td>cat:</td>
<td>pro2</td>
</tr>
<tr>
<td>noun:</td>
<td>see</td>
<td>noun:</td>
<td>see</td>
<td>noun:</td>
<td>see</td>
</tr>
<tr>
<td>verb:</td>
<td>see</td>
<td>verb:</td>
<td>see</td>
<td>verb:</td>
<td>see</td>
</tr>
<tr>
<td>prn:</td>
<td>3</td>
<td>prn:</td>
<td>3</td>
<td>prn:</td>
<td>3</td>
</tr>
</tbody>
</table>

The second hear mode operation to apply is PRD×OBJ (line 2 in 2.3.2). If the object is phrasal or clausal, PRD×OBJ adds the proplet of the initial surface, i.e., the determiner, preposition, or subordinating conjunction; then the proplets of the following surfaces require operations to complete the phrasal or clausal object. Here, however, the object is elementary and its addition requires no more than a single step which may be followed directly by S∪IP for adding the period (2.3.5).

2.3.4 Cross-copying see and you with PRD×OBJ (Line 2)

PRD×OBJ

### pattern level

- **cat:** #X
- **cat:** #X
- **arg:** Z
- **arg:** Z
- **fnc:** see
- **fnc:** see
- **verb:** β
- **verb:** β
- **noun:** α
- **noun:** α

### content level

<table>
<thead>
<tr>
<th>sur:</th>
<th>you</th>
<th>sur:</th>
<th>you</th>
<th>sur:</th>
<th>you</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat:</td>
<td>pro2</td>
<td>cat:</td>
<td>pro2</td>
<td>cat:</td>
<td>pro2</td>
</tr>
<tr>
<td>noun:</td>
<td>see</td>
<td>noun:</td>
<td>see</td>
<td>noun:</td>
<td>see</td>
</tr>
<tr>
<td>verb:</td>
<td>see</td>
<td>verb:</td>
<td>see</td>
<td>verb:</td>
<td>see</td>
</tr>
<tr>
<td>prn:</td>
<td>3</td>
<td>prn:</td>
<td>3</td>
<td>prn:</td>
<td>3</td>
</tr>
</tbody>
</table>

The #-marking of the nominative valency n' in the feature [cat: #n' a' v] of the first input proplet indicates that this valency position has been filled (2.3.3).

---

11 The agreement condition anticipates the analysis of the copula construction in Sect. 4.5.
The #-marking of the object valency position a’ in the feature [cat: #n’ #a’ v] of the first output proplet indicates that this valency position is now filled as well.

The hear mode derivation is completed with the addition of the period by standard S∪IP:

2.3.5 ABSORBING ⋅ INTO see WITH S∪IP (LINE 3)

For the complete set of proplets resulting from the hear mode derivation see 2.3.1.

Let us continue with point 4 on the to-do list 1.5.1, i.e. the DBS graph structure of the speak mode, based on the content derived in 2.3.2:

2.3.6 GRAPH ANALYSIS UNDERLYING PRODUCTION OF 2.3.1

(i) SRG (semantic relations graph)  (iii) NAG (numbered arcs graph)

(ii) signature

As in 2.1.5 and 2.2.9 the (iv) surface realization consists of three lines, showing the arc number, the surface realized from the goal proplet, and the traversal.

In accordance with point 5 of the to-list 1.5.1, let us list the steps of word form production. This format may be regarded as a non-graphical notational variant of the (iv) surface realization together with the (iii) NAG in 2.3.6.
2.3.7 SEQUENCE OF OPERATION NAMES AND SURFACE REALIZATIONS

arc 1: V/N from see to pro1

arc 2: N/V from pro1 to see

arc 3: V/N from see to pro2

arc 4: N/V from pro2 to see

The first to apply in the explicit sequence of speak mode operations is V/N:

2.3.8 Navigating with V/N from see to pro1 (arc 1)

V/N (1)

Based on the core value pro1, lexnoun in the sur slot of the output proplet realizes 1. The difference between an intransitive and a transitive verb is buffered by the variable X in the arg slot of the first proplet pattern: X is bound to NIL in 2.1.6, but to pro2 in 2.3.8.

To acquire the second continuation value in the arg slot of the verb, the navigation returns with the speak mode operation N/V:

2.3.9 Navigating with N/V from pro1 to see (arc 2)
The #-marking of the first arg value in the goal proplet resulted from the instruction of V\N (2.3.3). A #-marking instruction applies to a feature, here [arg: pro1 pro2], and not just to the value. For example, if a value in an arg slot is being #-marked, this does not affect the same value in a mdd slot.

Continuing the navigation from the predicate to the object is based on V\N:

2.3.10 Navigating with V\N from see to pro2 (arc 3)

\[
\begin{array}{c|c}
\text{pattern level} & \text{content level} \\
\hline
| verb: \alpha |\text{sur: lexnoun(}\beta\text{)} | \text{sur: you} | \\
| arg: #X \beta Y | noun: \beta | noun: pro2 | \\
| prn: K | fnc: \alpha | fnc: #see | \\
| & prn: K | & prn: 3 | \\
\end{array}
\]

The arg value pro1 is bound to the variable #X in the input pattern [arg: #X \beta Y], the arg value pro2 to \beta, and a possible third argument (in a three place verb) would be bound to Y.

The return from the object to the predicate with N\V is motivated by the need (i) to realize the punctuation mark (period), and (ii) to get into position for navigating to a next proposition (1.4.4):

2.3.11 Navigating with N\V from pro2 back to see (arc 4)

\[
\begin{array}{c|c}
\text{pattern level} & \text{content level} \\
\hline
| noun: \beta | \text{sur: lexverb(}\alpha\text{)} | \text{sur: •} | \\
| fnc: \alpha | verb: \alpha | verb: see | \\
| mdr: Z | arg: #X #\beta Y | arg: #n' #a' decl | \\
| prn: K | prn: K | prn: 3 | \\
| Z is #-marked or NIL | & & & \\
\end{array}
\]

The variable \alpha matches constants with or without a #-marking (1.6.5, 3). The #-marking of the second arg value resulted from the instruction of V\N.
For realizing the period, lexn noun uses the cat value decl in the goal proplet.

Extending DBS.2 to an example class with elementary subject/predicate/object required the definition of three operations: PRD × OBJ (h.22) for the hear mode, and V\N (s.3) and N\V (s.4) for the speak mode.

The hear mode derivation used three operation applications, while the speak mode derivation used four:

2.3.12 COMPARING HEAR AND SPEAK MODE SURFACE SEGMENTATION

hear mode: 1 2 3
           I × saw × you ∪ .

speak mode: 1 2 3 4
            V/N I N/V saw V\N you N\V.

In contradistinction to 2.2.15, the number of speak mode operations is higher than that of the hear mode. The reason is that the speak mode navigation must return to the top predicate, while connecting four words in the hear mode requires only three operation applications.

2.4 DBS.4: Phrasal Subject and Phrasal Object

In analogy to the upscaling from Sect. 2.1 (elementary subject and predicate) to Sect. 2.2 (phrasal subject and predicate), the current section upscales from Sect. 2.3 (elementary subject, predicate, and object) to a phrasal subject and a phrasal object. Following standard procedure, we begin with point 1 on the to-do list 1.5.1 by defining the content for the surface as a set of proplets:

2.4.1 CONTENT OF THE SURFACE The big dog saw a small bird.

The common nouns, here dog and bird, are absorbed into the determiners, the and a, respectively, resulting in the apparent order dog big and bird small.

Consider the time-linear hear mode derivation in canonical DBS format:
The proplets resulting from a hear mode derivation are in fact order-free.
The completed phrasal subject is connected to the predicate in line 3. The beginning of the phrasal object is connected to the predicate in line 4.

A phrasal noun preceding the verb, here *The big dog* (subject), is interpreted by the same operations as the phrasal noun following the verb, here *the small bird* (object), namely $\text{DET} \times \text{ADN}$ and $\text{DET} \cup \text{CN}$. This is based on omitting the verb feature in the operation patterns (compatibility by omission).

The sequence of explicit hear mode operation applications as they apply in 2.4.2 (point 3 on the to-do list 1.5.1) begins with $\text{DET} \times \text{ADN}$:

### 2.4.3 Cross-copying $n_1$ and big with $\text{DET} \times \text{ADN}$ (line 1)

The two input patterns of the operation accept any content proplets which match the features $[\text{noun: } N_n]$ and $[\text{adj: } \alpha]$. The difference between different determiners and between different adnominals is buffered by the variables. That determiner in line 2 has no $\text{fnc}$ value while the determiner in line 5 does is compensated by the absence of a $\text{verb}$ feature in the operation patterns.

The phrasal noun is completed with $\text{DET} \cup \text{CN}$ (2.2.4):
2.4.4 Absorbing dog into n_1 with DET\_CN (line 2)

\[
\text{DET\_CN (line 2)}
\]

In the concatenation of the...dog, conflict with a seemingly intervening proplet, here \textit{big}, is avoided by the way in which hear mode operations work. First, the core values of the determiner and \textit{big} are cross-copied (2.4.3). Then the common noun \textit{dog} is absorbed into the determiner and the two input proplets are replaced by one output proplet at the now front (2.4.4). The substitution of the variable serving as the core value of the determiner, here \textit{n_1}, applies not only to the determiner, but automatically and simultaneously to all instances of the variable in proplets, here \textit{big}, at the current now front.

The connection between the input proplets \textit{dog} and \textit{big} is coded in the mdr slot of \textit{dog} and the mdd slot of \textit{big} (2.4.3). The contribution of the determiner \textit{the} is coded by the \textit{sem} values \textit{def sg} (2.4.4). In short, the derivation of a phrasal subject results in a single proplet (2.2.5 2.4.5), no different for SBJ\_PRD than an elementary subject (2.1.2 2.3.3):

2.4.5 Cross-copying dog and saw with SBJ\_PRD (line 3)

\[
\text{SBJ\_PRD (line 3)}
\]
At this point, the hear mode derivation has interpreted The big dog saw.
The hear mode derivation continues with the beginning of the phrasal object,
using the \( \text{PRD} \times \text{OBJ} \) familiar from 2.3.4.

2.4.6 CROSS-COPYING see and a(n) with \( \text{PRD} \times \text{OBJ} \) (line 4)

\[
\begin{array}{cccc}
\text{PRD} \times \text{OBJ} & \text{pattern level} & \Rightarrow & \text{content level} \\
\verb: \beta & \text{cat: } \#X' \ N' \ Y \ \gamma & \text{noun: } \alpha & \text{sur: } \alpha \\
\arg: Z & \text{cat: } \text{CN}' \ N & \text{fnc: } \beta & \text{cat: } \text{CN}' \ N \\
\text{prn: } K & \text{arg: } Z & \text{prn: } K & \text{prn: } K \\
\end{array}
\]

The feature [cat: \#X' N' Y v] of the verb pattern ensures that the nominative
has already been concatenated (canceled valency position(s) \#X\[3\]) and that
there is a free valency position N' for another noun. The feature [cat: \text{CN'} N]
matches a determiner proplet beginning a phrasal object as well as an elementary
object proplet; in the latter case, CN' would be bound to NIL.

The next operation continues building the phrasal object with \( \text{DET} \times \text{ADN} \):

2.4.7 CROSS-COPYING \( n_2 \) AND small with \( \text{DET} \times \text{ADN} \) (line 5)

\[
\begin{array}{cccc}
\text{DET} \times \text{ADN} & \text{pattern level} & \Rightarrow & \text{content level} \\
\text{noun: } N_2 & \text{cat: } \text{CN}' \ NP & \text{adj: } \alpha & \text{sur: } \alpha \\
\text{cat: } \text{CN}' \ NP & \text{mdd: } \alpha & \text{cat: } \text{ADN} & \text{cat: } \text{CN}' \ NP \\
\text{mdd: } \alpha & \text{nc: } \text{nc} & \text{prn: } K & \text{prn: } K \\
\text{prn: } K & \text{sur: } \text{noun: } n_2 & \text{adj: small} & \text{adj: small} \\
\text{cat: } \text{sn'} \ \text{nsp} & \text{cat: } \text{sn'} \ \text{nsp} & \text{cat: } \text{small} & \text{cat: } \text{small} \\
\text{sem: } \text{indef sg} & \text{sem: } \text{indef sg} & \text{sem: } \text{small} & \text{sem: } \text{small} \\
\text{fnc: } \text{hit} & \text{fnc: } \text{hit} & \text{mdd: } \text{small} & \text{mdd: } \text{small} \\
\text{mdd: } \text{small} & \text{nc: } \text{nc} & \text{prn: } 4 & \text{prn: } 4 \\
\text{nc: } \text{nc} & \text{pc: } \text{pc} & \text{prn: } 4 & \text{prn: } 4 \\
\text{prn: } 4 & \text{prn: } 4 & \text{prn: } 4 & \text{prn: } 4 \\
\end{array}
\]

Here, the input proplet small matches the second input pattern of \( \text{DET} \times \text{ADN} \),
activating the operation. It happens to find the proplet \( n_2 \) at the now front
(n_1 has disappeared by absorption in 2.4.4 matching its first input pattern and applies. In the output, the core value of small is copied into the mdr slot of n_2, and the core value of n_2 is copied into the mdd slot of small.

The next word proplet bird matches the second input pattern of the operation DET∪CN, while its first pattern matches the n_2 proplets at the now front:

2.4.8 ABSORBING bird INTO n_2 WITH DET∪CN (line 6)

As shown in 2.4.2 line 6, simultaneous substitution replaces the variable n_2 not only in see but also in small (here omitted for reasons of space) with bird.

Finally, automatic word form recognition provides the period:

2.4.9 ABSORBING * INTO see WITH S∪IP (line 7)

We follow standard procedure and continue with point 4 on the to-do list 1.5.1 i.e. the canonical DBS graph analysis of the speak mode:

13 The underline ensures that #X′ is not bound to NIL. 1.6.5 (2).
2.4.10 Graph analysis underlying production of

(i) SRG (semantic relations graph)  
(ii) signature

(iii) NAG (numbered arcs graph)  
(iv) surface realization

The left branch with the arcs 1-4 resembles the graph 2.2.9 for an intransitive construction. The numbering of the NAG is called depth first in graph theory. It will be used throughout except for the gapping constructions in Sects. 5.2-5.4 and certain adverbial modifiers (3.2.11, 4.1.10, 4.2.8), for which a breadth first numbering is more natural (Sect. 1.4).

Following standard procedure, we continue with point 5 on the to-do list 1.5.1, i.e. the list of speak mode operation names with surface realizations:

2.4.11 Sequence of operation names and surface realizations

<table>
<thead>
<tr>
<th>Arc</th>
<th>Operation</th>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V/N</td>
<td>from see to dog</td>
<td>The</td>
</tr>
<tr>
<td>2</td>
<td>N\A</td>
<td>from dog to big</td>
<td>big</td>
</tr>
<tr>
<td>3</td>
<td>A\N</td>
<td>from big to dog</td>
<td>dog</td>
</tr>
<tr>
<td>4</td>
<td>N/V</td>
<td>from dog to see</td>
<td>saw</td>
</tr>
<tr>
<td>5</td>
<td>V\N</td>
<td>from see to bird</td>
<td>a</td>
</tr>
<tr>
<td>6</td>
<td>N\A</td>
<td>from bird to small</td>
<td>small</td>
</tr>
<tr>
<td>7</td>
<td>A\N</td>
<td>from small to bird</td>
<td>bird</td>
</tr>
<tr>
<td>8</td>
<td>N\V</td>
<td>from bird to see</td>
<td>.</td>
</tr>
</tbody>
</table>

In line with the DBS laboratory set-up, the proplets serving as input to the speak mode operations have been derived in the hear mode (2.4.3, 2.4.9). For navigating from one proplet to the next, the input pattern of a speak mode operation uses a continuation value and the prn value of the current proplet to find and activate a goal proplet.
Activating a next proplet in the agent's memory corresponds to retrieval in a conventional database. However, instead of retrieving an item for a user, autonomous navigation along the semantic relations between proplets stored in a word bank provides a selective activation of meaningful content. Goal proplets activated by navigation, in turn, may be mapped into external surfaces by the language-dependent lexicalization rules stored in the sur slot as the means for inter-agent language communication (automatic word form production).

The following speak mode operations 2.4.12–2.4.15 are analogous to 2.2.11–2.2.14 (the beginnings of the two examples are equivalent).

2.4.12 N\text{NAVIGATING WITH V/N FROM see TO dog} (arc 1)

\[ V/N (\text{\dag}) \]

\[
\begin{array}{|c|}
\hline
\text{pattern level} \\
\text{verb: } \alpha \\
\text{arg: } \beta \ X \\
\text{prn: K} \\
\hline
\end{array}
\Rightarrow
\begin{array}{|c|}
\hline
\text{content level} \\
\text{sur: } \uparrow \\
\text{verb: see} \\
\text{cat: } \#n' \ #a' \ \text{decl} \\
\text{sem: past} \\
\text{arg: dog bird} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: 4} \\
\hline
\end{array}
\begin{array}{|c|}
\hline
\text{sur: The} \\
\text{noun: dog} \\
\text{cat: snp} \\
\text{sem: def sg} \\
\text{fnc: see} \\
\text{mdr: big} \\
\text{nc: } \\
\text{pc: 4} \\
\hline
\end{array}
\]

#-mark $\beta$ in the arg slot of proplet $\alpha$

In the graph 2.4.10, V/N moves downward from V to N (arc 1) and realizes the determiner the from the N proplet dog.

To realize big, the navigation proceeds with N\text{\dag}A to the adnominal modifier:

2.4.13 N\text{NAVIGATING WITH N\dag A FROM dog TO big} (arc 2)

\[ N/\text{A (\dag)} \]

\[
\begin{array}{|c|}
\hline
\text{pattern level} \\
\text{noun: } \beta \\
\text{mdr: } \gamma \\
\text{prn: K} \\
\hline
\end{array}
\Rightarrow
\begin{array}{|c|}
\hline
\text{content level} \\
\text{sur: } \uparrow \\
\text{noun: dog} \\
\text{cat: snp} \\
\text{sem: def sg} \\
\text{fnc: see} \\
\text{mdr: big} \\
\text{nc: } \\
\text{pc: 4} \\
\hline
\end{array}
\begin{array}{|c|}
\hline
\text{sur: big} \\
\text{adj: big} \\
\text{cat: adn} \\
\text{sem: pad} \\
\text{mdd: dog} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: 4} \\
\hline
\end{array}
\]

#-mark $\gamma$ in the mdr slot of proplet $\beta$
The traversal sequence $V/N - N/A$ complies with the continuity principle of DBS (NLC 3.6.5), according to which a speak mode operation $Ax B$ may only be followed by an operation $BxC$ which accepts the output of $Ax B$ as input.

At this point, the navigation has realized *The big*. The only continuation feature currently available is $[\text{mdd: dog}]$:

2.4.14 **Navigating with $A \uparrow N$ from *big* back to *dog* (arc 3)**

In the graph 2.4.10, $A \uparrow N$ accesses the N proplet *dog* from below (arc 3) and realizes the common noun *dog*.

Now, the only continuation feature currently available is $[\text{fnc: see}]$:

2.4.15 **Navigating with $N/V$ from *dog* back to *see* (arc 4)**

In the graph 2.4.10, $N/V$ accesses the V proplet *see* from below (arc 4) and realizes the finite main verb *saw*.

The next step follows arc 5 to the phrasal object with $V\backslash N$, similar to 2.3.10.
2.4.16 **Navigating with $V \downarrow N$ from *see* to *bird* (arc 5)**

The different surface production of *you* in 2.3.10 and *a(n)* in 2.4.16 originates in the lexicalization rule lexnoun. It produces *you* from the feature [cat: sp2] in the output proplet of 2.3.10 and *a(n)* from the feature [sem: indef sg] in the output proplet of 2.4.16.

At this point, the speak mode derivation has realized The big dog saw the... The activated *bird* proplet matches the input proplet pattern of $N \downarrow A$.

2.4.17 **Navigating with $N \downarrow A$ from *bird* to *small* (arc 6)**

The operation (i) activates the *small* proplet in the word bank, and (ii) realizes the surface *small* from the core value with its lexicalization rule lexadj.

The activated *small* proplet matches the input pattern of the inverse operation $A \uparrow N$ to realize the final surface of discontinuous a...bird:
2.4.18 Navigating with $\mathbf{\wedge} N$ small back to bird (arc 7)

\[
\begin{array}{c}
\text{pattern level} \\
\text{adj: } \gamma \\
mdd: \beta \\
nc: Z \\
prn: K \\
\Rightarrow \\
\text{content level} \\
\text{sur: lexnoun}(\beta) \\
noun: \beta \\
mdr: \#\gamma \\
nc: Z \\
prn: K \\
\end{array}
\]

Z is NIL, or elementary and #-marked

$\uparrow$ $\downarrow$

The return of the navigation back to the verb with $N \backslash V$ is used to provide (i) the sur value. (period) with lexverb and to reach (ii) the unique location suitable for a possible navigation to a next proposition (1.4.4).

2.4.19 Navigating with $N \backslash V$ from bird back to see (arc 8)

\[
\begin{array}{c}
\text{pattern level} \\
noun: \beta \\
fnc: \alpha \\
mdr: Z \\
prn: K \\
\Rightarrow \\
\text{content level} \\
\text{sur: lexverb}(\alpha) \\
noun: \alpha \\
mdr: \#X \#\beta \# Y \\
nc: Z \\
prn: K \\
\end{array}
\]

The #-marking of the second arg value bird of the see proplet resulted from the instruction of $V \backslash N$ (2.4.16). It prevents lexnoun to realize bird again, but the variable Y would allow the realization of a third argument in a construction with a three-place (ditransitive) verb.

Applying DBS.4 to an example class with a phrasal subject and object did not require the definition of additional operations. The hear mode derivation used seven operation applications, while the speak mode derivation used eight:
2.4.20 Comparing Hear and Speak Mode Surface Segmentation

hear mode:

1 2 3 4 5 6 7

The × big ∪ dog × saw × a × small ∪ bird ∪ .

speak mode:

1 2 3 4 5 6 7 8

V/N the N↓A big A↑N dog N/V saw V\N a N↓A small A↑N bird N\V .

A DET-ADN-NOUN sequence requires two operation applications in a hear mode interpretation, but its speak mode production requires three.

2.5 DBS.5: Clausal Subject

Functor-argument relations are called intrapropositional if the argument is elementary (Sects. 2.1, 2.3) or phrasal (Sects. 2.2, 2.4). They are called extrapropositional, in contrast, if the argument is a subject, object, adnominal, or adverbial clause. Intra- and extrapropositional functor-argument relations are alike in that they are binary, i.e. defined by address between two individual, completely self-contained, order-free proplets. They differ in that the proplets of the former share a common prn value, while the latter use different prn values for the different component propositions.

This section complements the intrapropositional constructions of elementary (Sect. 2.1) and phrasal (Sect. 2.2) subjects with a clausal subject construction. Its matrix verb must (i) allow transitive use and (ii) be what is called a “psych verb” in lexicography, such as amaze, amuse, anger, annoy, bore, concern, excite, interest, irritate, satisfy, surprise, tire, and worry.

Let us follow standard procedure and begin with point 1 of 1.5.1 by presenting the example surface and its content as a set of concatenated proplets:

2.5.1 Content of the Surface That Fido barked amused Mary.

The subclause and the main clause have different prn values, here 5 and 6. The V/V relation between the two clauses is provided by (i) the additional
fnc attribute with the extrapropositional address value \textit{(amuse 6)} of the verb proplet \textit{bark} and (ii) the extrapropositional address value \textit{(bark 5)} in the first arg slot of the verb proplet \textit{amuse}.

While the previous examples used the sign kinds \textit{symbol}, e.g. \textit{dog}, and \textit{indexical}, e.g. \textit{you}, the current example uses the sign kind \textit{name} (CASM). The core value of a name is the referent, provided in an individual act of baptism rather than the lexicon of the language community (symbol) or some general pointing mechanism integrated into the agent’s onboard orientation (indexical). The names in 2.5.1 are accommodated by using the abstract referent addresses \textit{(dog x)} for \textit{Fido} and \textit{(person y)} for \textit{Mary}. Unlike proplets referring by means of concept matching (symbols) and pointing (indexicals), which lose their \textit{sur} value during a hear mode derivation, name proplets retain their \textit{sur} value in the form of a marker (FoCL 6.1.6), written in default font and lower case, e.g. \textit{fido} instead of \textit{Fido}. Name proplets are referred to by their marker in italics, e.g. \textit{fido}, in contradistinction to the surface value written in \textit{Helvetica}.

Consider the time-linear derivation of the content 2.5.1 as a hear mode graph:
In line 1, the v_1 proplet representing that and fido can not combine, because the agreement conditions between the lower subject fido and the lower predicate bark can not be checked. Therefore, the new operation subclause \(\sim\) SBJ relates the first two proplets by a suspension; it limits itself to assigning a prn value to the next word proplet.

When the missing verb bark arrives, the suspension is compensated by applying more than one operation in a single hear mode derivation step. Line 2a shows the combination of fido and bark with SBJ \(\times\) PRD by cross-copying, whereby the agreement conditions are taken into account. Line 2b shows the absorption of bark into v_1 with subclause \(\cup\) PRD.

Performing more than one operation in a derivation step is enabled by how the hear mode operations apply: (i) an operation is activated by a next word matching its second input pattern; (ii) an activated operation applies if it finds a proplet at the current now front matching its first input pattern. For example, SBJ \(\times\) PRD and subthat \(\times\) PRD are both activated by the next word proplet bark; SBJ \(\times\) PRD finds fido at the now front, while subthat \(\times\) PRD finds v_1.

Following standard procedure, we continue with point 3 on the to-do list 1.5.1. i.e. the sequence of explicit hear mode operations as they apply in the derivation 2.5.2, beginning with new subclause \(\sim\) SBJ.

---

14 A step of a hear mode derivation is defined as adding exactly one next word proplet to the now front.
2.5.3 Suspending \( v_1 \) and Fido with \( \text{sub}_{\text{clause}} \sim \text{SBJ} \) (line 1)

As a suspension, \( \text{sub}_{\text{clause}} \sim \text{SBJ} \) leaves the input proplets unchanged in the output, except for assigning the \( \text{prn} \) value of the subordinating conjunction to the lower subject. The subordinating conjunction \( v_1 \) is lexically defined (NLC A.4.4) with the attribute \( \text{arg} \), which is standard for a verb proplet, and an additional \( \text{fnc} \) attribute, which is usually reserved for nouns, but needed in a subordinating conjunction for the connection to the higher verb (2.5.6).

With the \( \text{fido} \) proplet present at the now front and the arrival of \( \text{bark} \) as the next word, standard \( \text{SBJ} \times \text{PRD} \) applies next:

2.5.4 Cross-copying \( \text{fido} \) and \( \text{barked} \) with \( \text{SBJ} \times \text{PRD} \) (line 2a)

Because the agreement and the cross-copying between the lower subject, here \( \text{dog x} \), and the lower verb, here \( \text{bark} \), have now been taken care of by \( \text{SBJ} \times \text{PRD} \), the new operation \( \text{sub}_{\text{clause}} \circ \text{PRD} \) may absorb the lower verb into the subordinating conjunction with complete grammatical detail:
2.5.5 Absorbing bark into v1 with sub\textit{clause} \uparrow PRD (line 2b)

The method of suspending a function word, here that in 2.5.3, until the proplet for it to fuse with arrives, here bark, is used also in the hear mode derivation of an object clause (2.6.7), adverbal clause (3.5.5), and adnominal clause repetition (5.5.10, 5.5.15).

The hear mode derivation continues by adding the next word amused as the higher verb with the new operation sub\textit{that} × PRD:

2.5.6 Cross-copying bark and amused with sub\textit{that} × PRD (line 3)

The variables K in the prn slot of the first input pattern and K+1 in the second output pattern assign an incremented prn value to the next word proplet, here 6 in the prn slot of amused. The cat pattern #X′ v of bark ensures that all valency
positions in the subclause have been canceled. The V/V connection between 
*bark* and *amuse* is established by cross-copying the core value of *bark* into the 
*arg* slot of *amuse* and the core value of *amuse* into the *fnc* slot of *bark*.

The object of *amuse* is *mary*. It is added with standard **PRD×OBJ**:

### 2.5.7 CROSS-COPYING *amuse* AND *Mary* WITH **PRD×OBJ** (line 4)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[verb: β]</td>
<td>sur: Mary</td>
</tr>
<tr>
<td>cat: #X’ VT</td>
<td>cat: noun (person y)</td>
</tr>
<tr>
<td>prn: K</td>
<td>cat: snp</td>
</tr>
<tr>
<td>arg: (bark 5)</td>
<td>sem: past</td>
</tr>
<tr>
<td></td>
<td>fnc: amuse</td>
</tr>
</tbody>
</table>

Agreement conditions as in [2.3.d]

It remains to connect the top verb and the punctuation mark, here . (period) 
with standard **S∪IP**:

### 2.5.8 ABSORBING . INTO *amused* WITH **S∪IP** (LINE 5)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat: #X’ VT</td>
<td>cat: verb: v.1</td>
</tr>
<tr>
<td>prn: K</td>
<td>prn:</td>
</tr>
<tr>
<td>arg: (bark 5)</td>
<td>sem: past</td>
</tr>
<tr>
<td></td>
<td>arg: (bark 5)(person y)</td>
</tr>
<tr>
<td>mdr:</td>
<td>mdr:</td>
</tr>
<tr>
<td>nc:</td>
<td>nc:</td>
</tr>
<tr>
<td>prn: 6</td>
<td>prn: 6</td>
</tr>
</tbody>
</table>

The proplets connected by this hear mode derivation have been stored by automatic word form recognition at the now front, with operations applying in situ (NLC Sect. 11.2). Clearing the now front by moving it into fresh territory (NLC Sect. 11.3) leaves the connected proplets behind in what has become an addition to the permanent storage area of the database, like sediment.
Following standard procedure, let us continue with point 4 on the to-do list 1.5.1, i.e., the canonical DBS graph structure of the speak mode, based on the content 2.5.1 resulting from the hear mode derivation 2.5.2.

2.5.9 Graph analysis underlying production of 2.5.1

(i) SRG (semantic relations graph)

(ii) signature

(iv) surface realization

The upper two branches resemble the transitive verb construction 2.3.6, while the lower branch resembles the intransitive verb construction 2.1.5.

The next item on the to-do list 1.5.1 is point 5, i.e., the list of speak mode operation names with surface realizations:

2.5.10 Sequence of operation names and surface realizations

Similar to the discontinuous phrasal noun structure The ... dog 2.2.10, the second surface of discontinuous That ... barked is realized on the return, here by N/V (arc 3). The extrapropositional subject/predicate transitions V/V 2.5.11 and V/V 2.5.14 resemble their intrapropositional counterparts V/N 2.1.6, 2.2.11, 2.3.8, 2.4.12, and N/V 2.1.7, e2.2.14, 2.3.11, 2.4.15 insofar as they are strictly binary, i.e., defined by address between two individual, completely self-contained, order-free proplets.
The next item on the to-do list is point 6, i.e. the sequence of explicit speak mode operations. The first operation to apply is new $V\langle V\rangle$. It navigates from the verb *amuse* of the matrix clause to the verb *bark* of the subject clause and realizes the conjunction *that*:

**2.5.11 Navigating with $V\langle V\rangle$ from *amuse* to *bark* (arc 1)**

\[ V\langle V\rangle \]

The continuation value enabling the application of $V\langle V\rangle$ is extrapropositional ($bark 5$) in the arg slot of *amuse*. The goal of the navigation is the verb *bark* representing the subclause. The relation is confirmed (and used for the return in arc 4, **2.5.14**) by the continuation value ($amuse 6$) in the fnc slot of *bark*.

Next the subclause is continued by adding the subject with standard $V\langle N\rangle$:

**2.5.12 Navigating with $V\langle N\rangle$ from *bark* to *fido* (arc 2)**

\[ V\langle N\rangle \]

The use of $V\langle N\rangle$ in a subclause is possible because the V pattern disregards the characteristic properties of a subclause verb, such as the presence of an
additional fnc feature and a subordinating conjunction, here that in the sem slot. Lexnoun uses the fido marker preserved in the sur slot of the fido proplet \(2.5.1\) for realizing the surface Fido.

The next speak mode operation is N/V for moving from fido to bark as the first step on the way back to the top predicate amuse and for realizing barked:

2.5.13 Navigating with N/V from fido back to bark (arc 3)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \quad \text{content level} \\
\text{noun: } \beta & \quad \text{sur: fido} \\
\text{fnc: } \alpha & \quad \text{sur: bark} \\
\text{mdr: } Z & \quad \text{noun: (dog x)} \\
\text{prn: } K & \quad \text{cat: snp} \\
\text{fnc: bark} & \quad \text{sem: nm m} \\
\text{mdr:} & \quad \text{fnc: (amuse 6)} \\
\text{nc:} & \quad \text{mdr:} \\
\text{pc:} & \quad \text{nc:} \\
\text{prn: 5} & \quad \text{prn: 5}
\end{align*}
\]

The second step needed for the return to the top predicate is based on the new operation V/V. It moves from the clausal subject represented by bark to the matrix verb amuse:

2.5.14 Navigating with V/V from bark back to amuse (arc 4)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \quad \text{content level} \\
\text{verb: } \beta & \quad \text{sur: bark} \\
\text{sem: that } Z & \quad \text{verb: amuse} \\
\text{arg: } \#X & \quad \text{cat: } \#n' \#a' \text{ decl +ip} \\
\text{fnc: (amuse 6)} & \quad \text{sem: past} \\
\text{mdr:} & \quad \text{arg: (bark 5) (person y)} \\
\text{nc:} & \quad \text{nc:} \\
\text{pc:} & \quad \text{pe:} \\
\text{prn: 5} & \quad \text{prn: 6}
\end{align*}
\]

By binding the extrapropositional address value (amuse 6) in the fnc slot of bark to the variable \((\alpha K+1)\) in the first pattern and using it as the core value of
the goal proplet, the operation navigates back to the top predicate. The relation is confirmed by the initial arg value (bark 5) of *amuse*.

The main clause is completed by navigating with standard V↓N to the grammatical object and realizing *Mary*:

### 2.5.15 Navigating with V↓N from *amuse* to *Mary* (arc 5)

\[
\begin{align*}
\text{pattern level} & \quad \text{content level} \\
& \quad \begin{array}{l}
\text{verb: } \alpha \\
\text{arg: } #X \beta \ Y \\
\text{prn: } K \\
\end{array} \quad \begin{array}{l}
\text{sur: } \text{lexnoun}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K \\
\end{array} \\
& \quad \begin{array}{l}
\text{sur: } \text{amuse} \\
\text{cat: } #n \ ' #a \ ' \ \text{decl} \\
\text{sem: } \text{past} \\
\text{arg: } #(\text{bark } 5) \ (\text{person } y) \\
\text{mdr: } Z \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 6 \\
\end{array} \quad \begin{array}{l}
\text{sur: } \text{Mary} \\
\text{noun: } (\text{person } y) \\
\text{cat: } \text{snp} \\
\text{sem: } \text{nm } f \\
\text{fnc: } \text{amuse} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 6 \\
\end{array} \\
& \quad \begin{array}{l}
\text{sur: } \text{amuse} \\
\text{cat: } #n \ ' #a \ ' \ \text{decl} \\
\text{sem: } \text{past} \\
\text{arg: } #(\text{bark } 5) \ #(\text{person } y) \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 6 \\
\end{array}
\end{align*}
\]

It remains to return to the top verb with N↓V to (i) provide the sur value (period) with lexverb and to (ii) reach the unique location suitable for a possible navigation to a next proposition by extrapropositional coordination (1.4.4).

### 2.5.16 Navigating with N↓V from *Mary* back to *amuse* (arc 6)

\[
\begin{align*}
\text{pattern level} & \quad \text{content level} \\
& \quad \begin{array}{l}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } \ Z \\
\text{prn: } K \\
\end{array} \quad \begin{array}{l}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{arg: } #X \ #\beta \ Y \\
\text{prn: } K \\
\end{array} \\
& \quad \begin{array}{l}
\text{sur: } \text{Mary} \\
\text{noun: } (\text{person } y) \\
\text{cat: } \text{snp} \\
\text{sem: } \text{nm } f \\
\text{fnc: } \text{amuse} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 6 \\
\end{array} \quad \begin{array}{l}
\text{sur: } * \\
\text{verb: } \text{amuse} \\
\text{cat: } #n \ ' #a \ ' \ \text{decl} \\
\text{sem: } \text{past} \\
\text{arg: } #(\text{bark } 5) \ #(\text{person } y) \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 6 \\
\end{array} \\
& \quad \begin{array}{l}
\text{sur: } \text{amuse} \\
\text{cat: } #n \ ' #a \ ' \ \text{decl} \\
\text{sem: } \text{past} \\
\text{arg: } #(\text{bark } 5) \ #(\text{person } y) \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 6 \\
\end{array}
\end{align*}
\]

Let us conclude the linguistic analysis of the example with point 7 on the to-do list [L.5.1]. Extending DBS.4 to an example class with a clausal subject required the definition of the three hear mode operations sub\text{clause}~SBJ (l[34]), sub\text{clause}∪PRD (l[10]), and sub\text{that}×PRD (l[15]), and the two speak mode operations V↓/V (s[5]) and V↓/V (s[6]). Counting the suspension compensation in line
The hear mode derivation used six operation applications, the same as the speak mode derivation:

**2.5.17 Comparing Hear and Speak Mode Surface Segmentation**

**Hear mode:**

1 2a 2b 3 4 5

That ∼ Fido × ∪ barked × amused × Mary ∪ .

**Speak mode:**

1 2 3 4 5 6

V/v Voice/Noun Fido N/v V/N barked V/v V/N amused V/v N/N Mary N/v V.

The 2a-2b sequence of ∼ and ∪ followed by barked shows the suspension compensation in the hear mode: barked is connected to Fido by cross-copying (∗) and absorbed into that (∪). The numbering of the connectives in the hear mode corresponds to the line numbering in 2.5.2.

### 2.6 DBS.6: Clausal Object

The higher verb of an object clause construction must be (i) transitive and (ii) in the class of what are called “mental state verbs” in lexicography, such as agree, believe, decide, forget, hear, know, learn, remember, regret, say, see, suspect, and want. The higher subject is the living (e.g. human) experiencer and the object is a clause which describes an event or other cognitive content inducing the mental experience named by the verb.

Let us follow standard procedure and begin with point 1 of 1.5.1 by defining the content for the example surface as a set of proplets concatenated by address:

**2.6.1 Content of the Surface** Mary heard that Fido barked.

<table>
<thead>
<tr>
<th>sur: mary</th>
<th>sur: hear</th>
<th>sur: fido</th>
<th>sur: bark</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: (person y)</td>
<td>verb: hear</td>
<td>noun: (dog x)</td>
<td>verb: bark</td>
</tr>
<tr>
<td>cat: snp</td>
<td>cat: #n #a decl</td>
<td>cat: snp</td>
<td>cat: #n decl</td>
</tr>
<tr>
<td>sem: nm f</td>
<td>sem: past</td>
<td>sem: nm m</td>
<td>sem: past</td>
</tr>
<tr>
<td>fnc: hear</td>
<td>arg: (person y) (bark 8)</td>
<td>fnc: bark</td>
<td>arg: (dog x)</td>
</tr>
<tr>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
</tr>
<tr>
<td>nc:</td>
<td>nc:</td>
<td>nc:</td>
<td>nc:</td>
</tr>
<tr>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
</tr>
<tr>
<td>prn: 7</td>
<td>prn: 7</td>
<td>prn: 8</td>
<td>prn: 8</td>
</tr>
</tbody>
</table>

In English declaratives, the different semantic roles of the subject and the object are expressed by their position: the subject precedes and the object follows...
the finite verb, regardless of whether they are elementary, phrasal, or clausal. In DBS, this is encoded by the arg value order of the predicate. For example, in 2.5.1 the order is \([\text{arg} : (\text{bark 5}) (\text{person y})]\) in \textit{amuse}, with (bark 5) representing the clausal subject, but in 2.6.1 it is \([\text{arg} : (\text{person y}) (\text{bark 8})]\) in \textit{hear}, with (bark 8) representing the clausal object.

Let us turn next to point 2 on the to-do list \[1.5.1\] i.e. the time-linear hear mode derivation of the example in canonical DBS graph format, to be followed by the sequence of explicit hear mode operation applications (point 3 of \[1.5.1\]), which serve as the declarative specification of the hear mode software.

2.6.2 \textit{Graphical hear mode derivation of the content} \[2.6.1\]

\begin{tabular}{|c|c|c|c|c|}
\hline
\textit{unanalyzed surface} & \textit{heard} & \textit{that} & \textit{Fido} & \textit{barked} \\
\hline
\textit{automatic word form recognition} & & & & \\
\hline
\begin{tabular}{l}
\textit{Mary} \textit{heard} \textit{that} \textit{Fido} \textit{barked} \\
\end{tabular} & \begin{tabular}{l}
\textit{sur: Mary} \textit{noun: (person y)} \\
\textit{cat: snp} \textit{sem: nm f} \textit{fnc: hear} \\
\end{tabular} & \begin{tabular}{l}
\textit{verb: hear} \textit{cat: #n’ a’ v} \textit{sem: past} \textit{arg: (person y)} \\
\textit{prn: 7} & \end{tabular} & \begin{tabular}{l}
\textit{verb: v_1} \textit{cat: #n’ a’ v} \textit{sem: that} \textit{arg: (person y)} \\
\textit{prn: 8} & \end{tabular} & \begin{tabular}{l}
\textit{verb: bark} \\
\textit{cat: snp} \textit{cat: n’ v} \textit{sem: past} \textit{arg: (person y)} \\
\textit{prn: 8} & \end{tabular} & \begin{tabular}{l}
\textit{verb: v_2} \textit{cat: v’ decl} \\
\textit{prn: 8} & \end{tabular} \\
\hline
\textit{syntactic-semantic parsing} & & & & \\
\hline
\begin{tabular}{l}
\textit{Mary} \textit{heard} \\
\end{tabular} & \begin{tabular}{l}
\textit{sur: Mary} \\
\textit{noun: (person y)} \\
\textit{cat: snp} \textit{sem: nm f} \textit{fnc: hear} \\
\textit{prn: 7} \end{tabular} & \begin{tabular}{l}
\textit{verb: hear} \textit{cat: #n’ a’ v} \textit{sem: past} \textit{arg: (person y)} \\
\textit{prn: 8} & \end{tabular} & \begin{tabular}{l}
\textit{verb: v_1} \textit{cat: #n’ a’ v} \textit{sem: that} \textit{arg: (person y)} \\
\textit{prn: 8} & \end{tabular} & \begin{tabular}{l}
\textit{verb: bark} \\
\textit{cat: snp} \textit{cat: n’ v} \textit{sem: past} \textit{arg: (person y)} \\
\textit{prn: 8} & \end{tabular} & \begin{tabular}{l}
\textit{verb: v_2} \textit{cat: v’ decl} \\
\textit{prn: 8} & \end{tabular} \\
\hline
\end{tabular}
There is a structural resemblance between the above derivation and the postverbal composition of a phrasal noun (2.4.2): the beginning of the object clause, i.e. that, is attached to the higher verb in line 2, after which the remainder of the object clause is added incrementally in lines 3 and 4 – similar to the incremental addition of a postverbal phrasal noun (except that a phrasal object is intrapropositional while a clausal object is extrapropositional).

The explicit operations sequence begins with standard SBJ×PRD:

2.6.3 CROSS-COPYING Mary AND heard WITH SBJ×PRD (line 1)

<table>
<thead>
<tr>
<th>SBJ×PRD (1)</th>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[noun: α]</td>
<td>verb: β</td>
<td>[noun: α]</td>
</tr>
<tr>
<td>cat: NP</td>
<td>cat: NP’ X v</td>
<td>cat: NP’ X v</td>
</tr>
<tr>
<td>fnc:</td>
<td>arg:</td>
<td>fnc:</td>
</tr>
<tr>
<td>prn: K</td>
<td>prn:</td>
<td>prn: K</td>
</tr>
</tbody>
</table>

Agreement conditions as in 2.1.2.

Once the subject and the predicate of the main clause are connected there follows the clausal object.
The first step in building the object clause is concatenating the main verb and the subordinating conjunction with PRD\times\text{sub}^{\text{that}}:

### 2.6.4 CROSS-COPYING hear and that with PRD\times\text{sub}^{\text{that}} (line 2)

\[
\text{PRD}\times\text{sub}^{\text{that}} (\text{line 2})
\]

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: \beta</td>
<td>sur: hear</td>
</tr>
<tr>
<td>cat: #NP' a' v</td>
<td>verb: v_1</td>
</tr>
<tr>
<td>arg: \gamma</td>
<td>sem: past</td>
</tr>
<tr>
<td>fnc:</td>
<td>arg: \gamma (V_n K+1)</td>
</tr>
<tr>
<td>prn: K</td>
<td>prn: K</td>
</tr>
<tr>
<td>\text{NP'} \epsilon {n', ns3', n-s3'}. \alpha \epsilon {\text{that, what}}. V \epsilon {v, vi}.</td>
<td></td>
</tr>
</tbody>
</table>

The variables K in the prn slot of the first input pattern and K+1 in the second output pattern assign an incremented prn value to the next word proplet, here 8 in the prn slot of v_1. One direction of the V\setminus V relation between hear and bark is established by copying the core value v_1 into the second arg slot of hear; the other direction is established by copying the core value of hear into the fnc slot of v_1. In [2.6.7] all occurrences of v_1 will be replaced by bark via simultaneous substitution.

Because the agreement between the lower subject fido and its verb cannot be dealt with before the arrival of bark, there is a suspension:

### 2.6.5 SUSPENDING v_1 AND Fido with sub^{\text{clause}}\sim\text{SBJ} (line 3)

\[
\text{sub}^{\text{clause}}\sim\text{SBJ} (\text{line 3})
\]

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: V_n</td>
<td>sur: Fido</td>
</tr>
<tr>
<td>sem: \alpha</td>
<td>verb: v_1</td>
</tr>
<tr>
<td>prn: K</td>
<td>cat: (dog x)</td>
</tr>
<tr>
<td></td>
<td>cat: snp</td>
</tr>
<tr>
<td>arg: \text{that}</td>
<td>arg: \text{that}</td>
</tr>
<tr>
<td>fnc: (hear 7)</td>
<td>fnc: (hear 7)</td>
</tr>
<tr>
<td>mdr:</td>
<td>mdr:</td>
</tr>
<tr>
<td>nc:</td>
<td>nc:</td>
</tr>
<tr>
<td>pc:</td>
<td>pc:</td>
</tr>
<tr>
<td>prn: 8</td>
<td>prn: 8</td>
</tr>
</tbody>
</table>

\alpha \epsilon \{\text{that, when, where, how, why}\}.
The operation leaves the input proplets unchanged, except for assigning the prn value 8 of the subordinating conjunction to the lower subject.

With the arrival of bark as the next word and the presence of the fido proplet at the now front, standard SBJ×PRD applies next:

2.6.6 Cross-copying fido and barked with SBJ×PRD (line 4a)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \text{content level} \\
\hline
\text{noun: } \alpha \text{ } & \text{verb: } \beta \text{ } & \text{noun: } \alpha \text{ } \\
\text{cat: NP} & \text{cat: NP' X v} & \text{cat: NP} \\
\text{fnc: } & \text{arg: } & \text{fnc: } \\
\text{prn: K} & \text{prn: } & \text{arg: } \\
\hline
\end{array}
\]

Agreement conditions as in 2.5.5.

In line 4b, an application of \( sub^{\text{clause}} \cup \text{PRD} \) compensates the suspension in line 3 by fusing the subordinating conjunction and the lower verb in the same way as in 2.5.5.

2.6.7 Absorbing barked into \( v_1 \) with \( sub^{\text{clause}} \cup \text{PRD} \) (line 4b)

\[
\begin{array}{c|c|c|c}
\text{pattern level} & \text{pattern level} & \text{content level} \\
\hline
\text{verb: } V'_{n} & \text{verb: } \beta & \text{verb: } \beta \\
\text{cat: } & \text{cat: NP' X v} & \text{cat: NP} \\
\text{sem: } Y & \text{arg: } & \text{sem: } Y \\
\text{arg: } & \text{arg: } & \text{arg: } \\
\text{prn: K} & \text{prn: K} & \text{prn: K} \\
\hline
\end{array}
\]

\( \alpha \in \{ \text{that, what, when, where, how, why} \} \)

It is possible to compensate the suspension in That Fido barked amused Mary (subject sentence) and Mary heard that Fido barked (object sentence).
with the same operation sub\textsuperscript{clause} ∪ PRD because the connection between the subordinating conjunction that and Fido is the same in both constructions\footnote{In the constructions exemplified by The dog which saw Mary (adnominal clause with subject gap) and The dog which Mary saw (adnominal clause with object gap), in contrast, the transitions which \times saw Mary and which Mary \times saw will require the different operations sub_{who} ∪ PRD (3.3.5) and sub_{whom} ∪ PRD (3.4.5), and will apply without a prior suspension.}

The hear mode derivation concludes with the application of S∪JP:

### 2.6.8 ABSORBING \* AND hear WITH S∪JP (LINE 5)

Following standard procedure, we continue with point 4 on the to-do list \ref{1.5.1}, i.e. the DBS graph structure of the speak mode, based on the content derived in the hear mode (2.6.2):

#### 2.6.9 GRAPH ANALYSIS UNDERLYING PRODUCTION OF 2.6.1

(i) SRG (semantic relations graph)  

\[
\begin{align*}
\text{hear} & \quad \text{mary} & \quad \text{bark} \\
\text{fido} &
\end{align*}
\]

(ii) signature

\[
\begin{align*}
N & \quad V & \quad N \\
\end{align*}
\]

(iii) NAG (numbered arcs graph)

\[
\begin{align*}
1 & : \text{hear} & 6 & : \\
2 & & 3 & : \\
4 & & 5 & : \\
\text{mary} & \quad \text{bark} & \quad \text{fido}
\end{align*}
\]

(iv) surface realization

\[
\begin{align*}
1 & : \text{Mary} & 2 & : \text{heard} & 3 & : \text{that} & 4 & : \text{Fido} & 5 & : \text{barked} & 6 & : \\
V/N & \quad N/V & \quad V/V & \quad V/N & \quad N/V & \quad V/V
\end{align*}
\]
The SRG, signature, and NAG differ from those of 2.5.9 in that the lower branch takes the object rather than the subject position of the higher clause.

The next item on the to-do list 1.5.1 is point 5, i.e. the list of speak mode operation names with surface realizations:

2.6.10 **SEQUENCE OF OPERATION NAMES AND SURFACE REALIZATIONS**

arc 1: \( V/N \) from hear to mary Mary 2.6.11
arc 2: \( N/V \) from mary to hear heard 2.6.12
arc 3: \( V/V \) from hear to bark that 2.6.13
arc 4: \( V/N \) from bark to fido Fido 2.6.14
arc 5: \( N/V \) from fido to bark barked 2.6.15
arc 6: \( V/V \) from bark to hear . 2.6.16

The matrix predicate hear may take an elementary, phrasal, or clausal object, just as the matrix predicate 

amuse in Sect. 2.5 may take an elementary, phrasal, or clausal subject. The extrapropositional object \( V/N \) (2.6.13) and \( V/V \) (2.6.16) resemble their intrapropositional counterparts \( V/V \) (2.3.10 2.4.16), and \( N/V \) (2.3.11 2.4.19) insofar as they are strictly binary, i.e. defined by address between two individual, completely self-contained, order-free proplets.

The next item on the to-do list 1.5.1 is point 6, i.e. the complete sequence of explicit speak mode operation applications, based on the proplets derived in the hear mode (2.6.3 2.6.8). The navigation begins with the traversal of arc 1 with standard \( V/N \):

2.6.11 **NAVIGATING WITH \( V/N \) FROM hear TO mary (arc 1)**

Using the continuation value hear of mary, the navigation returns to the predicate and realizes heard.
2.6.12 Navigating with N/V from mary back to hear (arc 2)

```
N/V (2)

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: β</td>
<td>sur: mary</td>
</tr>
<tr>
<td>fnc: α</td>
<td>noun: (person y)</td>
</tr>
<tr>
<td>mdr: Z</td>
<td>cat: snp</td>
</tr>
<tr>
<td>prn: K</td>
<td>sem: nn f</td>
</tr>
</tbody>
</table>

⇒ sur: lexverb(α)
verb: α
arg: #β X
prn: K

#-mark α in the fnc slot of proplet β

Z is NIL, or elementary and #-marked

⇑ ⇓

```

The condition is fulfilled because the input noun has no mdr (modifier) value, i.e. Z = NIL.

The next operation V\V is the object sentence counterpart to V/V in 2.5.11 for a subject sentence. That the higher predicate precedes in an English subject sentence (V/V) but follows in an object sentence (V\V) is coded in the respective arg patterns of the higher verb: in V/V it is [arg: (β K) X], but in V\V it is [arg: X (β K+1)].

2.6.13 Navigating with V\V from hear to bark (arc 3)

```
V\V (3)

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: α</td>
<td>sur: that</td>
</tr>
<tr>
<td>arg: X (β K+1)</td>
<td>verb: bark</td>
</tr>
<tr>
<td>prn: K</td>
<td>cat: #n' #a v</td>
</tr>
<tr>
<td></td>
<td>sem: that past</td>
</tr>
</tbody>
</table>

⇒ sur: lexverb(β)
verb: β
sem: that Z
fnc: (α K)
prn: K+1

#-mark β in the arg slot of proplet α

```

Having reached the predicate of the subclause and realized the subordinating conjunction, the navigation continues in the intrapropositional manner shown in Sects. 2.1, 2.4. The first step is moving from the predicate to the subject with standard V/v/N:

```
V/v/N

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: #hear 7</td>
<td>sur: that</td>
</tr>
<tr>
<td>fnc: hear</td>
<td>verb: bark</td>
</tr>
<tr>
<td>mdr:</td>
<td>cat: #n' v</td>
</tr>
<tr>
<td>sem: past</td>
<td>arg: (dog x)</td>
</tr>
<tr>
<td>arg: (person y) (bark 8)</td>
<td>fnc: #(hear 7)</td>
</tr>
<tr>
<td>mdr:</td>
<td>nc:</td>
</tr>
<tr>
<td>pc:</td>
<td>pc:</td>
</tr>
<tr>
<td>prn: 8</td>
<td>prn: 8</td>
</tr>
</tbody>
</table>
```

The next operation V\V is the object sentence counterpart to V/v/N in 2.5.11 for a subject sentence. That the higher predicate precedes in an English subject sentence (V/v/N) but follows in an object sentence (V\V) is coded in the respective arg patterns of the higher verb: in V/v/N it is [arg: X (β K+1)], but in V\V it is [arg: (β K) X].
2.6.14 NAVIGATING WITH V/N FROM bark TO (dog x) (arc 4)

\[
\begin{align*}
V/N (1) & \quad \Rightarrow \quad V/N (2)
\end{align*}
\]

The navigation is enabled by the continuation value (dog x) in the \textit{arg} slot of \textit{bark}.

Next, the navigation returns to the verb of the subclause to find the address of another valency filler or to continue to the top predicate.

2.6.15 NAVIGATING WITH N/V FROM Fido BACK TO bark (arc 5)

\[
\begin{align*}
N/V (2) & \quad \Rightarrow \quad N/V (3)
\end{align*}
\]

Here, the verb is one-place (intransitive). Therefore, the only option is the return to the matrix verb with V/V.

Given that \textit{hear} has no \textit{nc} value, V/V terminates the speak mode derivation with \textit{lexverb} realizing the period, based on the \textit{cat} value decl:
2.6.16 Navigating with V ∖ V from bark back to hear (arc 6)

Because the input pattern [arg: #X] of the subclause verb does not specify the number of valency positions, the operation is independent of the number of arguments in the subclause. It is sufficient for all the arg values to be #-marked. Thus, if the subclause verb were two- or three-place, V ∖ V would proceed from the subclause to the main clause in the same way as in 2.6.16 (after traversing the other valency fillers) – and similarly with V ∖ V in 2.5.14.

Let us conclude the linguistic analysis of the example with point 7 on the to-do list (1.5.1). Extending DBS.5 to an example class with a clausal object required the definition of the hear mode operation PRD × sub that (17), and the speak mode operations V ∖ V (8) and V ∖ V (9). Counting the suspension compensation in line 4b of 2.6.2, the hear mode derivation used six operation applications, the same as the speak mode derivation:

2.6.17 Comparing hear and speak mode surface segmentation

**hear mode:**

1 2 3 4a 4b 5

Mary × heard × that ∼ Fido × ⊖ barked ⊖ .

**speak mode:**

1 2 3 4 5 6

V/N Mary N/V heard V ∖ V that V/N Fido N/V bark V ∖ V .

The object clause with its suspension in the hear mode resembles the subject clause in 2.5.17.

---

16 A variable with a dotted underline matches values with or without a #-marking at the content level (1.6.5(3)).
3. Optional Functor-Arguments and Coordination

Modification and coordination have in common that they are optional. They differ in that the modifier-modified relation may connect proplets with different core attributes, e.g. adn|noun and adv|verb, while the conjunct—conjoint relation connects proplets with the same core attribute, i.e. noun—noun, verb—verb, or adn—adn.

3.1 DBS.7: Elementary Adverbial Modification

In English, the position of adverbial modifiers is comparatively free, while that of adnominal modifiers is fixed. More specifically, elementary adnominals are positioned between the determiner and the noun, while phrasal and clausal adnominals directly follow the modified (with extrapolation as an exception, CLaTR 9.3.5). Explicit DBS grammars for complex noun phrases with elementary adnominals have been defined in NLC Chaps. 13 (hear mode) and 14 (speak mode), and will not be repeated here. Instead this section will concentrate on multiple elementary adverbial modifiers.

Following standard procedure, we begin with point 1 on the to-do list 1.5.1 by defining the content for the example surface as a set of proplets:

3.1.1 CONTENT OF SURFACE Perhaps Fido is still sleeping there now.

---

1 The sem values mod(al), tmp (temporal), and loc(ational) of the elementary adverbials provide a first classification of their meaning. The value mod relates to the ancient distinction between the possible and the necessary in philosophy. In agent-based DBS, the distinction is reinterpreted as degrees of the agent’s certainty.
The adverbial modifiers occur in the positions before the finite auxiliary, between the auxiliary and the non-finite main verb, and following the main verb.

The order and kind of advs is not controlled by strict grammar rules, but by stylistic recommendations reflecting collocational conventions. These conventions may be recorded by the linguist and be introduced, to a degree, into the grammar software by the choice of the examples. The artificial agent using the grammar software may also record current use in the language community to further narrow the placement and choice of advs (learning, CLaTR Chap. 6).

The time-linear hear mode derivation of the content from the English surface 3.1.1 has the following DBS graph analysis:

3.1.2 Graphical hear mode derivation of the content 3.1.1

unanalyzed surface

Perhaps Fido is still sleeping there now

automatic word form recognition

syntactic−semantic parsing

suspension

cross−copying

cross−copying

Outside the DBS laboratory setup, the placement of the advs originates in the speak mode and is transmitted to the hearer by the incoming surfaces. Within the laboratory setup, the situation is reversed: the linguist provides a surface to the hear mode grammar which derives a content and the speak mode grammar uses that content to replicate the original surface.
There is no semantic relation between the first two word forms **Perhaps** and **Fido**. Therefore, the two are connected by a suspension operation called **ADV~NOM**. The second input pattern **NOM** matches the proplet representing **Fido**, while the first input pattern **ADV** matches the proplet representing **Perhaps**. As a suspension, **ADV~NOM** is limited to adding the **prn** value, here 9, to **fido**.

---

3. The **prn** value of initial word form in a text or an isolated linguistic example is provided by the parser.
3. Optional Functor-Arguments and Coordination

3.1.3 Suspending *perhaps* and Fido with ADV~NOM (line 1)

\[
\begin{align*}
\text{ADV~NOM} \quad (1) & \Rightarrow \quad [\text{adj: } \alpha] \quad [\text{noun: } \beta] \\
& \Rightarrow \quad [\text{adj: } \alpha] \quad [\text{noun: } \beta] \\
& \Rightarrow \quad [\text{adj: } \alpha] \quad [\text{noun: } \beta]
\end{align*}
\]

When \( v_1 \) is added in line 2, the suspension is compensated by applying two operations within the same step, namely SBJ x PRD for \( fido \times v_1 \) and ADV x FV for \( perhaps \times v_1 \). Both operations are activated by the current next word \( v_1 \) matching their second input pattern and both succeed by finding proplets at the now front which match their respective first input pattern:

3.1.4 Cross-copying *fido* and *is* with SBJ x PRD (line 2a)

\[
\begin{align*}
\text{SBJ x PRD} \quad (\text{II}) & \Rightarrow \quad [\text{noun: } \alpha] \quad [\text{verb: } \beta] \\
& \Rightarrow \quad [\text{noun: } \alpha] \quad [\text{verb: } \beta]
\end{align*}
\]

3.1.5 Cross-copying *perhaps* and \( v_1 \) with ADV x FV (line 2b)

\[
\begin{align*}
\text{ADV x FV} \quad (\text{III}) & \Rightarrow \quad [\text{adj: } \alpha] \quad [\text{verb: } \gamma] \\
& \Rightarrow \quad [\text{adj: } \alpha] \quad [\text{verb: } \gamma]
\end{align*}
\]
Following the auxiliary, another adverbial is added by new FV×ADV:

### 3.1.6 CROSS-COPYING v_1 AND still WITH FV×ADV (line 3)

**FV×ADV (line 3)**

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: γ</td>
<td>sur: still</td>
</tr>
<tr>
<td>mdr: X</td>
<td>adj: still</td>
</tr>
<tr>
<td>prn: K</td>
<td>verb: v_1</td>
</tr>
<tr>
<td></td>
<td>cat: adv</td>
</tr>
<tr>
<td></td>
<td>sem: pres</td>
</tr>
<tr>
<td></td>
<td>arg: (dog x)</td>
</tr>
<tr>
<td></td>
<td>prn: 9</td>
</tr>
<tr>
<td></td>
<td>mdr: perhaps</td>
</tr>
<tr>
<td></td>
<td>prn: K</td>
</tr>
<tr>
<td>ADV ∈ {adv, adv'v}</td>
<td>sur: sleeping</td>
</tr>
<tr>
<td></td>
<td>verb: sleep</td>
</tr>
<tr>
<td></td>
<td>cat: be</td>
</tr>
<tr>
<td></td>
<td>sem: pres</td>
</tr>
<tr>
<td></td>
<td>arg: (dog x)</td>
</tr>
<tr>
<td></td>
<td>prn: 9</td>
</tr>
<tr>
<td></td>
<td>mdr: perhaps</td>
</tr>
<tr>
<td></td>
<td>prn: K</td>
</tr>
</tbody>
</table>

With the finite verb in place, FV×ADV applies without suspension.

In accordance with the DBS laboratory set-up, the hear mode must prepare for the speak mode by coding the information needed for a correct surface realization. To enable lexverb to apply a second time for realizing the separate non-finite part of the verb, FV×ADV has introduced the separator “!” before the postverbal values in the mdr slot of the finite auxiliary ([mdr: X ! α]).

Next AUX∪NFV (2.2.7) fuses is and sleeping:

**AUX∪NFV (line 4)**

<table>
<thead>
<tr>
<th>pattern level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: V_n</td>
</tr>
<tr>
<td>cat: #W’ AUX’ VT</td>
</tr>
<tr>
<td>sem: X</td>
</tr>
<tr>
<td>prn: K</td>
</tr>
<tr>
<td>Agreement conditions omitted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>sur: sleeping</td>
</tr>
<tr>
<td>verb: sleep</td>
</tr>
<tr>
<td>cat: be</td>
</tr>
<tr>
<td>sem: pres</td>
</tr>
<tr>
<td>arg: (dog x)</td>
</tr>
<tr>
<td>prn: 9</td>
</tr>
<tr>
<td>mdr: perhaps</td>
</tr>
<tr>
<td>prn: K</td>
</tr>
</tbody>
</table>

In lines 5 and 6, FV×ADV (3.1.6) reapplies to add two more adverbials.

---

4 For the speak mode, however, an adverbial intervening between the finite auxiliary and the non-finite main verb presents a challenge insofar as it creates the discontinuous structure is...sleeping. This is in contradistinction to lexverb realizing a phrasal verb in one single speak mode step, as in 2.2.14. The discontinuous structure at hand resembles the “Satzklammer” in German (3.3.4).
3.1.8 **Cross-copying sleep and there with FV x ADV** (line 5)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \\
\text{verb: } \gamma & \text{sur: there} & \text{sur: sleep} \\
\text{mdr: X} & \text{adj: there} & \text{cat: sleep} \\
\text{prn: K} & \text{cat: adv} & \text{sem: pres} \\
\text{adj: } \alpha & \text{sem: loc} & \text{arg: (dog x)} \\
\text{mdd: } \alpha & \text{mdr: } X \alpha & \text{mdr: perhaps! still there} \\
\text{prn: } \gamma & \text{prn: } 9 & \text{prn: } 9 \\
\end{array}
\]

The derivation is completed by adding the period with **SUBP**:

3.1.9 **Cross-copying sleep and now with FV x ADV** (line 6)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \\
\text{verb: } \gamma & \text{sur: now} & \text{sur: sleep} \\
\text{mdr: X} & \text{adj: now} & \text{cat: sleep} \\
\text{prn: K} & \text{cat: adv} & \text{sem: pres} \\
\text{adj: } \alpha & \text{sem: loc} & \text{arg: (dog x)} \\
\text{mdd: } \alpha & \text{mdr: } X \alpha & \text{mdr: perhaps! still there now} \\
\text{prn: } \gamma & \text{prn: } 9 & \text{prn: } 9 \\
\end{array}
\]

3.1.10 **Absorbing * into sleep with SUBP** (line 7)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \\
\text{verb: } \beta & \text{sur: *} & \text{sur: sleep} \\
\text{cat: } \#X VT & \text{verb: V_1} & \text{cat: sleep} \\
\text{prn: K} & \text{cat: } \#X SM ip+ & \text{sem: pres prog} \\
\text{Agreement conditions as in 1.3.2} & \text{arg: (dog x)} & \text{arg: (dog x)} \\
\text{mdr: perhaps! still there now} & \text{mdr: perhaps! still there now} & \\
\text{prn: } 9 & \text{prn: } 9 & \\
\end{array}
\]

Let us follow standard procedure and continue with point 4 on the to-do list 1.5.1, i.e. the DBS graph showing the semantic relations of structure and the surface realization in the speak mode:
3.1.11 **Graph Analysis Underlying Production of**

(i) **SRG (semantic relations graph)**

(ii) **signature**

(iii) **NAG (numbered arcs graph)**

(iv) **surface realization**

In this example, neither the depth first nor the breadth first arc numbering (Sect. 1.4) provides a consecutive numbering.

We continue with point 5 on the to-do list 1.5.1, i.e. the list of speak mode operation names with surface realizations:

3.1.12 **Sequence of Operation Names and Surface Realizations**

<table>
<thead>
<tr>
<th>Arc</th>
<th>Operation</th>
<th>From</th>
<th>To</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>V↓A</td>
<td>sleep</td>
<td>perhaps</td>
<td>5.1.13</td>
</tr>
<tr>
<td>4</td>
<td>A↑V</td>
<td>perhaps</td>
<td>sleep</td>
<td>3.1.14</td>
</tr>
<tr>
<td>1</td>
<td>V/N</td>
<td>sleep</td>
<td>fido</td>
<td>3.1.15</td>
</tr>
<tr>
<td>2</td>
<td>N/V</td>
<td>fido</td>
<td>sleep</td>
<td>3.1.16</td>
</tr>
<tr>
<td>5</td>
<td>V↓A</td>
<td>sleep</td>
<td>still</td>
<td>3.1.17</td>
</tr>
<tr>
<td>6</td>
<td>A↑V</td>
<td>still</td>
<td>sleeping</td>
<td>3.1.18</td>
</tr>
<tr>
<td>7</td>
<td>V↓A</td>
<td>sleep</td>
<td>there</td>
<td>3.1.19</td>
</tr>
<tr>
<td>8</td>
<td>A↑V</td>
<td>there</td>
<td>sleep</td>
<td>3.1.20</td>
</tr>
<tr>
<td>9</td>
<td>V↓A</td>
<td>sleep</td>
<td>now</td>
<td>3.1.21</td>
</tr>
<tr>
<td>10</td>
<td>A↑V</td>
<td>now</td>
<td>sleep</td>
<td>3.1.22</td>
</tr>
</tbody>
</table>

Each downward traversal is directly followed by the associated upward traversal. The semantic relations used by the navigation were established by the time-linear hear mode derivation 3.1.2.

The next item on the to-do list 1.5.1 is point 6, i.e. the sequence of explicit speak mode operations. The first operation to apply is V↓A. It navigates from the verb *sleep* to the adv *bark* and realizes preverbal *Perhaps*:

---

5 The ! marker is inserted only once when FV×ADV applies for the first time.
3. Optional Functor-Arguments and Coordination

3.1.13 Navigating with $V \downarrow A$ from *sleep* to *perhaps* (arc 3)

$V \downarrow A$ (9)

```
| verb: $\alpha$ | sur: lexadj($\beta$) |
| mdr: #X $\beta$ $Y$ | adj: $\beta$ |
| prn: K | mdd: $\alpha$ |
```

$\Rightarrow$

```
| sur: Perhaps |
| adj: perhaps |
| cat: adv |
| mdd: sleep |
| mdr: perhaps ! still there now |
| nc: |
| pc: |
| prn: 9 |
```

In this traversal of an initial adverbial, the value assigned to #X is NIL. The instruction $\#$-marks the first unmarked mdr value, i.e. initial *perhaps*, to prevent another traversal. Lexadj realizes the surface from the goal proplet.

The navigation returns with $A \uparrow V$ to the verb for a continuation value:

3.1.14 Navigating with $A \uparrow V$ from *perhaps* back to *sleep* (arc 4)

$A \uparrow V$ (10)

```
| adj: $\beta$ |
| mdd: $\alpha$ |
| prn: K |
```

$\Rightarrow$

```
| sur: lexverb($\hat{\alpha}$) |
| verb: $\alpha$ |
| mdr: X $\beta$ $Y$ |
| prn: K |
```

The uncanceled nominative, i.e. (dog x), prevents lexverb from realizing (any part of) the verb.

The *sleep* proplet offers two continuation values, (dog x) for *fido* in the arg slot and still in the mdr slot. The ! in the mdr slot prevents a navigation to still; also, more than one adverbial preceding the finite verb would deviate from realizing the original input surface and is against the collocational conventions recorded by the agent. Therefore, the other continuation value (dog x) is accessed with $V \downarrow N$:
3.1.15 Navigating with V/N from sleep to fido (arc 1)

\[ \begin{array}{|c|c|} \hline \text{pattern level} & \text{content level} \\ \hline \verb|verb: \alpha| & \text{sur: lexnoun(} \beta \text{)} \\ \verb|arg: \beta X| & \text{noun: } \beta \\ \verb|prn: K| & \text{fnc: } \alpha \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|} \hline \text{sur: Fido} & \text{verb: sleep} \\ \text{noun: (dog x)} & \text{cat: np} \\ \text{cat: #ns3' #be'} & \text{sem: def sg} \\ \text{sem: pres prog} & \text{fnc: sleep} \\ \text{arg: (dog x)} & \text{mdr: #perhaps ! still there now} \\ \text{nc:} & \text{nc:} \\ \text{pc:} & \text{pc:} \\ \text{prn: 9} & \text{prn: 9} \\ \hline \end{array} \]

The instruction # marks the first (and only) arg value of sleep (see the goal proplet in 3.1.16) to prevent another realization of the subject.

At this point, the speak mode derivation has realized Perhaps Fido. The only continuation value of (dog x) is sleep:

3.1.16 Navigating with N/V from (dog x) back to sleep (arc 2)

\[ \begin{array}{|c|c|} \hline \text{pattern level} & \text{content level} \\ \hline \verb|noun: \beta| & \text{sur: fido} \\ \verb|fnc: \alpha| & \text{verb: sleep} \\ \verb|mdr: Z| & \text{cat: np} \\ \verb|prn: K| & \text{sem: def sg} \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|} \hline \text{sur: is} & \text{verb: is} \\ \text{noun: (dog x)} & \text{cat: #ns3' #be'} \\ \text{cat: #ns3' #be'} & \text{sem: pres prog} \\ \text{sem: pres prog} & \text{arg: # (dog x)} \\ \text{arg: # (dog x)} & \text{mdr: #perhaps ! still there now} \\ \text{nc:} & \text{nc:} \\ \text{pc:} & \text{pc:} \\ \text{prn: 9} & \text{prn: 9} \\ \hline \end{array} \]

The condition of N/V ensures that the noun is either elementary (here (dog x) is not modified) or the modifier has already been traversed (e.g. 2.2.14). Based on the ! separator in the mdr slot sleep, lexverb realizes only the finite auxiliary is, using the #ns3' and #be' values stored in the cat slot and the pres value in the sem slot of sleep.

Having decoupled the finite auxiliary and the non-finite main verb by realizing only is, the navigation continues with V/A to the next adverbial, i.e. the unmarked value still following the separator in the mdr slot of sleep.
3. Optional Functor-Arguments and Coordination

3.1.17 Navigating with V↓A from sleep to still (arc 5)

The example shows why the #-marking applies to continuation values, and not to the core value: the core value is unique, but there may be several continuation values to be canceled one after the other. As shown by 2.5.14, #-marking may apply to continuation values not only in start but also in goal proplets.

The only continuation value of still is sleep. Using A↑V, the navigation returns from still to sleep.

3.1.18 Navigating with A↑V from still back to sleep (arc 6)

Because squeezing another adverbial between the finite auxiliary and the non-finite main verb would deviate from realizing the original input surface and be against the collocational conventions recorded by the agent, lexverb realizes sleeping using the core value and the prog value in the sem slot.

---

6 Thus, the speak mode realizes the non-finite part of the verb in a transition from an adverbial and not from the finite auxiliary. This is in fulfillment of the Continuity Condition (NLC 3.6.5) in 3.1.17 and 3.1.18.
At this point, the speak mode derivation has realized Perhaps Fido is still sleeping. It remains to add the two adverbials following the non-finite main verb, beginning with there:

3.1.19 Navigating with $V \downarrow A$ from sleep to there (arc 7)

\[
\begin{array}{c}
\text{pattern level} \\
\text{content level}
\end{array}
\]

\[
\begin{array}{c}
\text{sur: lexadj(β)} \\
\text{verb: sleep} \\
\text{cat: #n’ #be’ decl} \\
\text{sem: pres prog} \\
\text{arg: #(dog x)} \\
\text{mdr: #perhaps ! #still there now} \\
\text{nc:} \\
\text{prn: 9}
\end{array}
\Rightarrow
\begin{array}{c}
\text{sur: lexadj(β)} \\
\text{adj: β} \\
\text{mdd: α} \\
\text{prn: K}
\end{array}
\]

The result of the instruction is shown in 3.1.20 as the #-marking of there in the mdr slot of sleep.

At this point, the navigation has realized Perhaps Fido is still sleeping there, with there as the last activated proplet. The next step is a return to the verb with $A \uparrow V$ to (i) find another adverb as a continuation value, to (ii) add the interpunctuation sign, or to (iii) continue to another proposition.

3.1.20 Navigating with $A \uparrow V$ from there back to sleep (arc 8)

\[
\begin{array}{c}
\text{pattern level} \\
\text{content level}
\end{array}
\]

\[
\begin{array}{c}
\text{sur: lexverb(α)} \\
\text{verb: sleep} \\
\text{cat: #n’ #be’ decl} \\
\text{sem: pres prog} \\
\text{arg: #(dog x)} \\
\text{mdr: #perhaps ! #still #there now} \\
\text{nc:} \\
\text{prn: 9}
\end{array}
\Rightarrow
\begin{array}{c}
\text{sur: lexverb(α)} \\
\text{adj: β} \\
\text{mdd: α} \\
\text{prn: K}
\end{array}
\]

In the present case, the mdr slot of sleep still has the adverbial now as a value without #-marking. Therefore $V \downarrow A$ applies once more to navigate to the adverbial for realization:
3.1.21 **Navi gat ing with $\mathcal{V}_A$ from *sleep* to *now* (arc 9)**

\[
\begin{align*}
\text{pattern level} & \\
\text{verb: } \alpha & \\
\text{mdd: } X & \\
\text{prn: } K & \\
\Rightarrow & \\
\text{content level} & \\
\text{sur: } \text{lexadj}(\hat{\beta}) & \\
\text{adj: } \beta & \\
\text{mdd: } \alpha & \\
\text{prn: } K & \\
\end{align*}
\]

#-mark $\beta$ in the mdr slot of proplet $\alpha$

The result of the instruction is shown in 3.1.22 as the #-marking of *now* in the mdr slot of *sleep*.

Another return to the verb completes the speak mode derivation:

3.1.22 **Navi gat ing with $A^\uparrow V$ from *now* back to *sleep* (arc 10)**

\[
\begin{align*}
\text{pattern level} & \\
\text{adj: } \beta & \\
\text{mdd: } \alpha & \\
\text{prn: } K & \\
\Rightarrow & \\
\text{content level} & \\
\text{sur: } \text{lexverb}(\hat{\alpha}) & \\
\text{verb: } \alpha & \\
\text{mdd: } X & \\
\text{prn: } K & \\
\end{align*}
\]

Without any untraversed continuation values in the *sleep* proplet, lexverb uses the cat value decl to realize the period.

Following standard procedure, we conclude the section with point 7 on the to-do list [1.5.1]. Extending DBS.6 to an example class with four elementary adverbial modifiers required the definition of the three hear mode operations $\text{ADV}\neg\text{NOM}$ (h32), $\text{ADV}\times\text{FV}$ (h2), and $\text{FV}\times\text{ADV}$ (h43), and the two speak mode operations $\mathcal{V}_A$ (s9) and $A^\uparrow V$ (s10).

Counting the suspension compensation in line 2b, the hear mode derivation used eight operation applications, while the speak mode derivation used ten:
3.1.23 Comparing Hear and Speak Mode Surface Segmentation

**Hear mode:**
```
1  2a  2b  3  4  5  6  7
Perhaps ~ Fido × × is × still ∪ sleeping × there × now ∪ .
```

**Speak mode:**
```
3  4  1  2  5  6
V↓A Perhaps A↑V V/N Fido N/V is V↓A still A↑V sleeping
7  8  9  10
V↓A there A↑V V↓A now A↑V .
```

The numbering of the speak mode operations corresponds to the arcs traversed in 3.1.11. The additional operation applications in production are caused by the four adverbials, which each require two traversals in the speak mode, but only one concatenation in the hear mode.

3.2 DBS.8: Phrasal Adverbial Modification

While English may restrict the adnominal vs. adverbial use of elementary modifiers morphologically, as in beautiful (adnominal) vs. beautifully (adverbial), no such morphological restrictions hold for phrasal modifiers like on the table. In adnominal use phrasal modifiers must follow the modified noun, while the positioning of the adverbial use is relatively free, similar to the elementary adverbials illustrated in the previous section.

If a prepositional phrase is positioned after the verb and a noun (e.g. following the object position), there is an ambiguity between the adnominal and the adverbial use. For example, Fido ate the bone on the table may mean that the bone is on table (adnominal) or that the eating is on the table (adverbial).

In DBS, the different readings are coded as the following contents:

3.2.1 Adnom. Content of Surface Fido ate the bone on the table.

```
noun: (dog x)  cat: snp  cat: #n' #a' v  cat: adnv snp  cat: adnv snp
sem: nm f  sem: past  sem: def sg  sem: on def sg
fnc: eat  arg: (dog x) bone  fnc: eat  mdd: bone  <
mdr:  mdr: table  mdr: bone  mdr: bone
nc:  nc:  nc:  nc:  nc:
pv:  prn: 11  prn: 11  prn: 11  prn: 11
```

Here, the mdd value of table is bone and the mdr value of bone is table.
3.2.2 Adverbial Content of Surface

Fido ate the bone on the table.

Here the mdd value of table is eat and the mdr value of eat is table.

The adverbial reading has the following hear mode graph analysis:

3.2.3 Graphical Hear Mode Derivation of the Content

```plaintext

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: (dog x)</td>
<td>verb: eat</td>
<td>noun: bone</td>
<td>cat: snp</td>
<td>cat:</td>
<td>noun: (dog x)</td>
</tr>
<tr>
<td>cat: snp</td>
<td>cat: #n' #a' v</td>
<td>cat: snp</td>
<td>cat:</td>
<td>cat:</td>
<td>cat: snp</td>
</tr>
<tr>
<td>sem: nm f</td>
<td>sem: past</td>
<td>sem: def</td>
<td>sem:</td>
<td>sem:</td>
<td>sem: on</td>
</tr>
<tr>
<td>fnc: eat</td>
<td>arg: (dog x) bone</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
</tr>
<tr>
<td>mdr:</td>
<td>table</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
</tr>
<tr>
<td>nc:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>prn: 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Here the mdd value of table is eat and the mdr value of eat is table.

The adverbial reading has the following hear mode graph analysis:

3.2.3 Graphical Hear Mode Derivation of the Content

```

```
The initial declarative sentence continues with a prepositional phrase which is attached to the verb via cross-copying into the respective mdr and mdd slots in line 4. Thereby the preposition on matches the second input pattern of the operation $\text{PRD} \times \text{PREP}^{\text{on}}$ (3.2.7), which finds the verb eat at the now front matching its first input pattern. The apparently intervening noun bone is no obstacle because proplets at the now front are retrieved by pattern, and not by surface position (i.e. surface adjacency is not required).

The hear mode derivation begins with an application of standard $\text{SBJ} \times \text{PRD}$:

### 3.2.4 CROSS-COPYING fido and ate WITH $\text{SBJ} \times \text{PRD}$ (line 1)

<table>
<thead>
<tr>
<th>SBJ×PRD (1)</th>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat: np</td>
<td>arg: α prn: K</td>
<td>prn: 10</td>
</tr>
<tr>
<td>prn: 10</td>
<td>prn: 10</td>
<td>prn: 10</td>
</tr>
</tbody>
</table>

Agreement conditions as in 2.1.2.
The second hear mode operation to apply is standard PRD×OBJ:

3.2.5 CROSS-COPYING eat AND the WITH PRD×OBJ (line 2)

The underline of the variable #X' ensures that there is at least one #-marked counterpart in the matching input (1.6.5, 2).

The grammatical object is completed with standard DET∪CN:

3.2.6 ABSORBING bone INTO n_1 WITH DET∪CN (line 3)

At this point, the hear mode derivation has interpreted the beginning of the example up to Fido ate the bone.

The next operation is called PRD×PREPm. In this name, the PRD part (for predicate) indicates adverbial use, the × connective indicates cross-copying, and PREPm the use of a prepositional phrases as an optional modifier.

7 The operation PRD×PREPm (4.1.6) is also adverbial, but cancels a valency position in the verb as an obligatory argument.
3.2.7 CROSS-COPYING eat AND on WITH PRD×PREP⁰ (line 4)

The adverbial modifier relation between the prepositional phrase and the finite verb is established by cross-copying the core value of the preposition, i.e. n_2, into the mdr slot of the verb eat and the core value of eat into the mdd slot of the preposition. The cat value adnv indicates that a prepositional phrase may be used adnominally (adn) or adverbially (adv). The lexical entry uses the English surface as the initial value of the sem attribute to specify the kind of preposition, here on in contradistinction to, e.g., in, under, below, etc.

The first step in assembling the remainder of the prepositional phrase (CLaTR 7.2.5) is based on the operation PREP∪NP:

3.2.8 ABSORBING the INTO n_2 WITH PREP∪NP (line 5)

Function word absorption in combination with simultaneous substitution automatically replaces all occurrences of the variable n_2 (originating lexically in the preposition) with the incremented variable n_3 (originating lexically in the determiner). Also, the cat values and the sem values of the two input proplets are combined in the output proplet.
The time-linear assembling of the prepositional phrase is completed with the application of $\text{DET} \cup \text{CN}$.

### 3.2.9 Absorbing table into $n_3$ with $\text{DET} \cup \text{CN}$ (line 6)

There is no grammatical upper limit on repeating the addition of prepositional phrases. Here, however, the hear mode derivation is completed with the application of standard $\text{S} \cup \text{IP}$ for adding the period:

### 3.2.10 Absorbing * into eat with $\text{S} \cup \text{IP}$ (line 7)

This concludes point 3 on the to-list i.e. the sequence of explicit hear mode operations deriving 3.2.1.

---

8 The initial adnv value in the cat slot of the first input proplet originated lexically in the preposition (3.2.7, 3.2.8). It is accommodated by the initial cat variable $X$ restricted to adnv or NIL (2.1.3).

9 As an example see on the table under the tree in the garden in Sect. 5.1, the semantic relation between the stacked items is modification: under the tree modifies table and in the garden modifies tree. There is also a semantically less restricted relation, namely coordination, as in for two days in the water without a life vest (NLC, Sects. 15.2, 15.3).
The next point is the DBS graph structure of the speak mode, based on the content derived in the hear mode (3.2.3):

3.2.11 Graph analysis underlying production of 3.2.2

(i) SRG (semantic relations graph)

(ii) signature

(iii) NAG (numbered arcs graph)

Based on the breadth first arc numbering (Sect. 1.4), the transition numbers in the surface realization are consecutive.

Point 5 on the to-do list 1.5.1 is the list of speak mode operations with surface realizations:

3.2.12 Sequence of operation names and surface realizations

<table>
<thead>
<tr>
<th>Arc</th>
<th>Operation</th>
<th>Surface Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V/N from eat to fido</td>
<td>Fido</td>
</tr>
<tr>
<td>2</td>
<td>N/V from fido to eat</td>
<td>ate</td>
</tr>
<tr>
<td>3</td>
<td>V\N from eat to bone</td>
<td>the_bone</td>
</tr>
<tr>
<td>4</td>
<td>N\V from bone to eat</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>V↓N from eat to table</td>
<td>on_the_table</td>
</tr>
<tr>
<td>6</td>
<td>N↑V from table to eat</td>
<td></td>
</tr>
</tbody>
</table>

The navigation accesses the top predicate, here eat, from an extrapositional coordination (1.4.4) and continues with standard V/N to the initial value in its arg slot to realize Fido (arc 1). From there it returns with N/V from fido to eat to realize ate (arc 2). The initial declarative sentence is concluded with the V\N traversal and realization of the object (arc 3), and the empty return to the predicate (arc 4). From there, the adverbial modifier phrase is accessed via V↓N and realized (arc 5). The final return via N↑V to the predicate (arc 6) realizes the period and gets into position for traversing a next proposition.
3.2.13 **Navigating with V/\N from eat to dog** (arc 1)

\[
\begin{array}{c}
\text{pattern level} \\
\text{content level}
\end{array}
\]

\[
\begin{array}{c}
\text{verb: } \alpha \\
\text{arg: } \beta X \\
\text{prn: } K
\end{array}
\Rightarrow
\begin{array}{c}
\text{sur: lexnoun(}\hat{\beta}\text{)} \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K
\end{array}
\]

#-mark \( \hat{\beta} \) in the arg slot of proplet \( \alpha \)

Lexnoun realizes the surface *Fido* from the marker “fido,” which the hear mode operation 3.2.4 stored in the sur slot of the name proplet. The name surface covers the name marker temporarily until the surface is moved to the agent’s action component for word form realization, revealing the marker again.

3.2.14 **Navigating with N/V from dog back to eat** (arc 2)

\[
\begin{array}{c}
\text{pattern level} \\
\text{content level}
\end{array}
\]

\[
\begin{array}{c}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K
\end{array}
\Rightarrow
\begin{array}{c}
\text{sur: lexverb(}\hat{\alpha}\text{)} \\
\text{verb: } \alpha \\
\text{arg: } \#\hat{\beta} X \\
\text{prn: } K
\end{array}
\]

#-mark \( \hat{\alpha} \) in the fnc slot of proplet \( \beta \)

Z is NIL, or elementary and #-marked

At this point, the speak mode has realized *Fido ate*.

The transitive sentence beginning the example with the realization of the subject and the predicate continues with the traversal of the grammatical object, navigating from *eat* to *bone* and back, based on the standard \( \V/\N \) and \( \N/\V \) operations. While the \( \V/\N \) traversal realizes the _bone_, the \( \N/\V \) traversal is empty:
### 3.2.15 Navigating with \( \mathcal{V} \backslash \mathcal{N} \) from *eat* to *bone* (arc 3)

The \#-marking of the valency positions in the *cat* slot of *eat* stem from the hear mode operation patterns \( \text{SBJ} \times \text{PRD} \) (3.2.4) and \( \text{PRD} \times \text{OBJ} \) (3.2.5). The \#-marking of the initial *arg* value (*dog x*), in contrast, stems from the instruction of the speak mode operation 3.2.13.

### 3.2.16 Navigating with \( \mathcal{N} \backslash \mathcal{V} \) from *bone* back to *eat* (arc 4)

The \#-marking of second *arg* value *bone* of *eat* resulted from the instruction in 3.2.15. Lexverb in the *sur* slot of the goal proplet is prevented from realizing an interpunctuation sign by the presence of the unmarked continuation value *table* in the *mdr* slot of *eat*.

The navigation continues with the traversal of the phrasal adverbial modifier. This requires the new operation \( \mathcal{V} \downarrow \mathcal{N} \). The modification relation is indicated by the connective \(|\). The adverbial nature of the modification is indicated by the source node \( \mathcal{V} \). The phrasal nature of the modifier is indicated by the goal node \( \mathcal{N} \):
3. Optional Functor-Arguments and Coordination

3.2.17 Navigating with \( \text{V ↓ N} \) from *eat* to *table* (arc 5)

### Pattern Level

\[
\begin{array}{c}
\text{verb: } \alpha \\
\text{mdd: } X \beta Y \\
\text{prn: } K
\end{array}
\Rightarrow
\begin{array}{c}
\text{sur: xlnoun(} \hat{\beta} \text{)} \\
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{array}
\]

- \( \# \)-mark in the mdr slot of proplet \( \alpha \)

### Content Level

\[
\begin{array}{c}
\text{sur: on_the_table} \\
\text{verb: eat} \\
\text{cat: #ns3' #a' decl} \\
\text{sem: past} \\
\text{arg: #(dog x) #bone} \\
\text{mdr: table} \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 10}
\end{array}
\]

The realization of the surface *on_the_table* by lexverb is based on the sem value *on*, the sem values *def* sg, and the core value *table*.

Using the mdd value of the goal proplet, the navigation returns to the verb with \( \text{N ↑ V} \). This provides the following options: If the mdr slot of the goal proplet *eat* contained a value without \( \# \)-marking, the navigation could continue with another prepositional phrase \(^{10}\) If the nc slot of the goal proplet contained a verb value, the navigation could continue with an intra- or extrapropositional coordination.

The third option, which is taken here, is to complete the traversal of the content \(^{3.2.2}\) without any untraversed continuation values in the *eat* proplet, lexverb uses the cat value decl to realize the period:

3.2.18 Navigating with \( \text{N ↑ V} \) from *table* back to *eat* (arc 6)

### Pattern Level

\[
\begin{array}{c}
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{array}
\Rightarrow
\begin{array}{c}
\text{sur: lexverb(} \hat{\alpha} \text{)} \\
\text{verb: } \alpha \\
\text{mdd: } X \beta Y \\
\text{prn: } K
\end{array}
\]

### Content Level

\[
\begin{array}{c}
\text{sur: } \\
\text{verb: eat} \\
\text{cat: #ns3' #be' decl} \\
\text{sem: past} \\
\text{arg: #(dog x) #bone} \\
\text{mdr: #table} \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 10}
\end{array}
\]

---

\(^{10}\) Just as there is no upper limit on repeating the addition of prepositional phrases in the hear mode, there is no upper limit in the speak mode either (NLC 15.3.8).
Let us conclude the linguistic analysis of the example with point 7 on the to-do list. Extending DBS.7 to a class of examples with a phrasal adverbial modifier in postverbal position required the definition of four additional operations, namely \( \text{PRD} \times \text{PREP} \) (125) and \( \text{PREP} \cup \text{NP} \) (138) for the hear mode, and \( \text{V} \downarrow \text{N} \) (113) and \( \text{N} \uparrow \text{V} \) (14) for the speak mode. The hear mode derivation used seven operation applications, while the speak mode derivation used six:

### 3.2.19 Comparing Hear and Speak Mode Surface Segmentation

**Hear mode:**

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\text{Fido} \times \text{ate} \times \text{the} \cup \text{bone} \times \text{on} \cup \text{the} \cup \text{table} \cup .
\end{array}
\]

**Speak mode:**

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
\text{V/N Fido} \text{N/V ate} \text{V/N the_bone} \text{V/N the_table} \text{A/V} \uparrow \text{V} .
\end{array}
\]

The numbering of the speak mode operations corresponds to the breadth first arc numbering in 3.2.11. The hear mode derivation runs straight through without any suspension; the speak mode derivation has one empty traversal.

### 3.3 DBS.9: Clausal Adnominal Modifier with Subject Gap

Having analyzed an example with an elementary (Sect. 3.1) and a phrasal (Sect. 3.2) modifier, let us turn to clausal modifiers. We begin with adnominal, aka relative, clauses. They may occur with a subject gap, as in *The dog which saw Mary barked*, and with an object gap, as in *The dog which Mary saw barked*. Consider the subject gap example and its content:

#### 3.3.1 Content of the Surface

*The dog which saw Mary barked.*

The order-free proplets of this content are shown in the order resulting from their hear mode derivation 3.3.2. The extrapropositional nature of the construction is reflected by the use of two prn values, 12 and 13.
3. Optional Functor-Arguments and Coordination

One direction of the relation between the subclause and the main clause is coded by the feature \(\text{mdr: (see 13)}\) of \textit{dog}. The other direction is coded by the feature \(\text{mdd: (dog 12)}\) value of \textit{see}. The subject gap is shown by the feature \(\text{arg: } \emptyset \text{(person y)}\) of \textit{see}; the implicit filler of the gap is shown by the feature \(\text{mdd: (dog 12)}\) directly underneath.

Let us turn next to point 2 on the to-do list \([1.5.1]\) i.e. the time-linear hear mode derivation of the example in canonical DBS graph format:

### 3.3.2 Graphical hear mode derivation of the content 3.3.1

<table>
<thead>
<tr>
<th>The</th>
<th>dog</th>
<th>which</th>
<th>saw</th>
<th>Mary</th>
<th>barked</th>
</tr>
</thead>
</table>
| **unanalyzed surface**
| sur: the dog | noun: n_1 | cat: sn | sem: def | fnc: | mdr: prn: |
| sur: which | verb: v_1 | cat: n’ a’ v | sem: which | arg: 0 | mdd: prn: |
| sur: saw | verb: see | cat: sn | sem: which | fnc: | prn: |
| sur: Mary | noun: (person y) | cat: snp | sem: past | fnc: see | prn: |
| sur: barked | noun: (person y) | cat: n’ v | sem: past | fnc: | prn: |
| sur: | verb: v_2 | cat: v’ decl | |

**automatic word form recognition**

1. sur: the dog
   - noun: n_1
   - cat: sn
   - sem: def
   - fnc:
   - mdr: prn: 12

2. sur: which
   - verb: v_1
   - cat: n’ a’ v
   - sem: which
   - arg: 0
   - mdd: prn: 13

3. sur: saw
   - verb: see
   - cat: sn
   - sem: which
   - arg: 0
   - mdd: (dog 12)
   - prn: 13

4. sur: Mary
   - noun: (person y)
   - cat: snp
   - sem: past
   - arg: 0
   - mdd: (dog 12)
   - prn: 13

5. sur: barked
   - noun: (person y)
   - cat: n’ v
   - sem: past
   - arg: see
   - mdd: (dog 12)
   - prn: 13

**syntactic–semantic parsing**

- absorption
- cross-copying
- absorption with simultaneous substitution
- cross-copying
- cross-copying
The addition of the subordinating conjunction which in line 2 leaves it open whether the adnominal clause will have a subject or an object gap. This is because which has no morphological case marking, unlike the who/whom alternation. The issue is decided in line 3 with the arrival of the verb. It is absorbed into the subordinating conjunction and the gap indicator ∅ in the cat slot of which is moved from lexically neutral to subject position (cf. line 4).

The adnominal clause is completed in line 4 with the object Mary. In line 5, the head noun dog of the adnominal clause is connected to the predicate bark of the matrix clause by cross-copying their core values into the respective fnc and arg slots. In short, the adnominal clause represented by see is connected to dog by the modifier|modified relation (3.3.4) and the head noun dog is related to bark by the subject/predicate relation (3.3.9).

Let us continue with point 3 on the to-do list beginning with DET∪CN:

3.3.3 ABSORBING dog INTO The with DET∪CN (line 1)

Next, the new operation CN×subwho connects dog and lexical v_1 (which) by cross-copying the core values into the respective mdr and mdd slots:
3. Optional Functor-Arguments and Coordination

3.3.4 CROSS-COPYING dog AND which WITH CN×subwho (line 2)

CN×subwho attaches an adnominal clause with a subject gap, as in man who saw Mary, while CN×subwhom attaches an adnominal clause with an object gap, as in man whom Mary saw.

CN×subwho and CN×subwhom are alike in that they increment the prn value of the subordinating conjunction; they differ in their case assignment. If the conjunction which is used (instead of who vs. whom), there is a brief ambiguity because CN×subwho and CN×subwhom apply both.

The new operation subwho PRD absorbs see into the conjunction which:

3.3.5 ABSORBING see INTO v_1 WITH subwho PRD (line 3)
In [3.3.4] \( \emptyset \) was moved by \( \text{CN}\times \text{sub}^{\text{who}} \) from its lexical middle position in the \text{arg} slot of \text{which} to subject position, fixing the interpretation to a clausal adnominal modifier with a subject gap.

The \#-marking of the \( n' \) valency position in the initial \text{cat} slot of \text{see} enables the application of standard \( \text{PRD}\times \text{OBJ} \). The operation combines the \text{see} proplet resulting from the previous derivation step and the next word proplet \text{mary} by cross-copying the core values into the respective \text{arg} and \text{fnc} slots:

### 3.3.6 Cross-copying see and Mary with \( \text{PRD}\times \text{OBJ} \) (line 4)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{PRD}\times \text{OBJ} ) (\text{line 4})</td>
<td></td>
</tr>
<tr>
<td>\verb: ( \beta )</td>
<td></td>
</tr>
<tr>
<td>\text{cat:} ( #X'\ N' \ Y \gamma )</td>
<td></td>
</tr>
<tr>
<td>\text{arg:} ( Z )</td>
<td></td>
</tr>
<tr>
<td>\text{fnc:} ( Z \alpha )</td>
<td></td>
</tr>
<tr>
<td>\text{prn:} ( K )</td>
<td></td>
</tr>
<tr>
<td>\text{noun:} ( \alpha )</td>
<td></td>
</tr>
<tr>
<td>\text{cat:} ( \text{CN}' \ N )</td>
<td></td>
</tr>
<tr>
<td>\text{prn:} ( K )</td>
<td></td>
</tr>
<tr>
<td>Agreement conditions as in [3.3.4]</td>
<td></td>
</tr>
<tr>
<td>sur: Mary</td>
<td></td>
</tr>
<tr>
<td>\text{verb:} \text{see}</td>
<td></td>
</tr>
<tr>
<td>\text{cat:} ( #n' \ a' \ v )</td>
<td></td>
</tr>
<tr>
<td>\text{sem:} \text{which past}</td>
<td></td>
</tr>
<tr>
<td>\text{arg:} ( \emptyset )</td>
<td></td>
</tr>
<tr>
<td>\text{mdd:} ( \text{dog 12} )</td>
<td></td>
</tr>
<tr>
<td>\text{nc:}</td>
<td></td>
</tr>
<tr>
<td>\text{pc:}</td>
<td></td>
</tr>
<tr>
<td>\text{prn:} ( 13 )</td>
<td></td>
</tr>
<tr>
<td>sur: ( \text{Mary} )</td>
<td></td>
</tr>
<tr>
<td>\text{noun:} ( \text{person y} )</td>
<td></td>
</tr>
<tr>
<td>\text{cat:} \text{snp}</td>
<td></td>
</tr>
<tr>
<td>\text{sem:} \text{nm f}</td>
<td></td>
</tr>
<tr>
<td>\text{fnc:} \text{mdr:}</td>
<td></td>
</tr>
<tr>
<td>\text{nc:}</td>
<td></td>
</tr>
<tr>
<td>\text{pc:}</td>
<td></td>
</tr>
<tr>
<td>\text{prn:} ( 13 )</td>
<td></td>
</tr>
<tr>
<td>sur: ( \text{mary} )</td>
<td></td>
</tr>
<tr>
<td>\text{verb:} \text{see}</td>
<td></td>
</tr>
<tr>
<td>\text{cat:} ( #n' \ a' \ v )</td>
<td></td>
</tr>
<tr>
<td>\text{sem:} \text{which past}</td>
<td></td>
</tr>
<tr>
<td>\text{arg:} ( \emptyset ) ( \text{person y} )</td>
<td></td>
</tr>
<tr>
<td>\text{mdd:} ( \text{dog 12} )</td>
<td></td>
</tr>
<tr>
<td>\text{nc:}</td>
<td></td>
</tr>
<tr>
<td>\text{pc:}</td>
<td></td>
</tr>
<tr>
<td>\text{prn:} ( 13 )</td>
<td></td>
</tr>
</tbody>
</table>

If the subclause verb were one-place, as in \text{The dog which snored amused Mary}, \( \text{PRD}\times \text{OBJ} \) could not apply at this point. If the subclause verb were three-place, as in \text{The dog which gave Suzy a look amused Mary}, \( \text{PRD}\times \text{OBJ} \) would apply once more. In short, after adding the subordinating conjunction (“relative pronoun”) to the head noun (3.3.4), there are no restrictions on subsequent continuations within the subclause.

The next word \text{barked} matches the second input pattern \( \text{SBJ}\times \text{PRD} \), which finds \text{fid} at the now front matching its first input pattern and applies:.

### 3.3.7 Cross-copying dog and barked with \( \text{SBJ}\times \text{PRD} \) (line 5)

<table>
<thead>
<tr>
<th>Pattern Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{SBJ}\times \text{PRD} ) (\text{line 5})</td>
</tr>
<tr>
<td>\verb: ( \beta )</td>
</tr>
<tr>
<td>\text{cat:} ( #N' \ X \ v )</td>
</tr>
<tr>
<td>\text{arg:} ( \alpha )</td>
</tr>
<tr>
<td>\text{fnc:} ( K )</td>
</tr>
<tr>
<td>\text{prn:} ( K )</td>
</tr>
<tr>
<td>\text{noun:} ( \alpha )</td>
</tr>
<tr>
<td>\text{cat:} \text{NP}</td>
</tr>
<tr>
<td>\text{fnc:} ( K )</td>
</tr>
<tr>
<td>\text{prn:} ( K )</td>
</tr>
</tbody>
</table>

\( ^{11} \) To handle the alternation between, e.g. the dog which barks and the dogs which bark, the range \( \{\text{NP}'\} \) of valency positions matching the \text{cat} value of, e.g. \text{dog}, are coded into the \text{cat} slot of \text{which}. For simplicity, the special agreement with auxiliaries is omitted.
Like an elementary or phrasal noun, the dog which saw Mary may serve not only as subject, but also as direct object, e.g., Bill fed the dog which saw Mary, or as indirect object, e.g., Bill gave the dog which saw Mary to Suzy.

The hear mode derivation is completed with an application of \( S \cup IP \):

### 3.3.8 ABSORBING INTO bark WITH \( S \cup IP \) (line 6)

Using the cat value decl of the lexical punctuation sign, the output of \( S \cup IP \) characterizes the top verb bark as a declarative.

Following standard procedure, we continue with point 4 of 1.5.1:

### 3.3.9 GRAPH ANALYSIS UNDERLYING PRODUCTION OF 3.3.1

(i) SRG (semantic relations graph) (ii) signature (iii) NAG (numbered arcs graph)
(iv) surface realization

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
\text{The\_dog} & \text{which\_saw} & \text{Mary} & \text{barked\_} \\
V/N & N|V & V/N & N/V & N/V \\
\end{array}
\]

The modifier role of the relative clause is shown graphically by the V|N line between see and dog. The subject gap is shown by the missing subject/predicate branch below see (NLC 7.3.1). The arc numbering is depth first.

In accordance with point 5 on the to-do list 1.5.1, the word form production may be summarized as the following list of speak mode operations:

3.3.10 Sequence of Operation Names and Surface Realizations

<table>
<thead>
<tr>
<th>Arc</th>
<th>Operation</th>
<th>Surface</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V/N</td>
<td>from bark to dog</td>
<td>The_dog</td>
</tr>
<tr>
<td>2</td>
<td>N↓V</td>
<td>from dog to see</td>
<td>which saw</td>
</tr>
<tr>
<td>3</td>
<td>V_N</td>
<td>from see to mary</td>
<td>Mary</td>
</tr>
<tr>
<td>4</td>
<td>N_V</td>
<td>from mary to see</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>V↑N</td>
<td>from see to dog</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>N/V</td>
<td>from dog to bark</td>
<td>barked_</td>
</tr>
</tbody>
</table>

The navigation begins with V/N, traveling from the top verb to the subject and realizing its surface. For a V/N application it makes no difference whether or not the goal proplet serves as the modified of a clausal adnominal modifier. This is because the semantic relations between proplets are strictly binary (1.2.1, 1.2.2), such that an N/V traversal with V/N (arc 1) is unaffected by a subsequent V|N traversal with N\_V (arc 2).

Let us turn to the next item on the to-do list 1.5.1 i.e. the sequence of explicit speak mode operations. The first traversal moves with V/N from the top predicate to the top subject. Its role as the head noun of an adnominal clause, coded by the feature [mdr: (see 13)], is ignored by the pattern of V/N:

3.3.11 Navigating with V/N from bark to dog (arc 1)
The lexicalization rule lexnoun in the sur slot of the output pattern realizes the surface \textit{The dog} from the core and sem values of the output proplet.

The next operation is new $N \downarrow V$, which has $V \uparrow N$ (3.3.15) as its counterpart. Both use the semantic relation of modification and connect the adnominal subclause extrapropositionally with the head noun in the main clause:

### 3.3.12 Navigating with $N \downarrow V$ from dog to see (arc 2)

\[
\begin{array}{c}
\text{pattern level} \\
\begin{bmatrix}
\text{nou}: \alpha \\
\text{mdr}: (\beta K+1) \\
\text{prn}: K
\end{bmatrix} \\
\uparrow \\
\begin{bmatrix}
\text{sur}: \text{lexverb}(\hat{\beta}) \\
\text{verb}: \beta \\
\text{mdd}: (\alpha K) \\
\text{prn}: K+1
\end{bmatrix}
\Rightarrow \\
\begin{bmatrix}
\text{sur}: \text{which}_\text{saw} \\
\text{verb}: \text{see} \\
\text{cat}: \#n \#a' v \\
\text{sem}: \text{which past} \\
\text{arg}: \emptyset (\text{person y}) \\
\text{mdd}: (\text{dog 12}) \\
\text{nc}: \\
\text{pc}: \\
\text{prn}: 13
\end{bmatrix}
\end{array}
\]

The vertical matching and binding between the mdr variable ($\beta K+1$) and the continuation value (see 13) is a prerequisite for the operation $N \downarrow V$ to apply successfully. The same holds for the incremented prn value 13, which was assigned by the hear mode operation $CN \times \text{sub}^{\text{who}}$ in 3.3.4 and is now required by the patterns [mdr: ($\beta K+1$)] and [prn: K+1]. Using the features [verb: see], [sem: which past], and [arg: $\emptyset$ (person y)], lexverb realizes which_saw.

After reaching the verb of the adnominal subclause, the navigation travels with intrapropositional $V \downarrow N$ to the object and realizes the surface Mary:

### 3.3.13 Navigating with $V \downarrow N$ from see to mary (arc 3)

\[
\begin{array}{c}
\text{pattern level} \\
\begin{bmatrix}
\text{verb}: \alpha \\
\text{arg}: \#X \beta Y \\
\text{prn}: K
\end{bmatrix} \\
\uparrow \\
\begin{bmatrix}
\text{sur}: \text{lexnoun}(\hat{\beta}) \\
\text{nou}: \beta \\
\text{fnc}: \alpha \\
\text{prn}: K
\end{bmatrix}
\Rightarrow \\
\begin{bmatrix}
\text{sur}: \text{Mary} \\
\text{nou}: (\text{person y}) \\
\text{cat}: \text{snp} \\
\text{sem}: \text{nm f} \\
\text{fnc}: \text{see} \\
\text{mdr}: \\
\text{nc}: \\
\text{pc}: \\
\text{prn}: 13
\end{bmatrix}
\end{array}
\]
To complete the traversal of the adnominal subclause, standard $N\downarrow V$ returns from the lower object *mary* to the lower predicate *see*.

### 3.3.14 Navigating with $N\downarrow V$ from *mary* back to *see* (arc 4)

The traversal is “empty” in the sense that no surface is produced by lexverb.

The next navigation step uses the new $V\uparrow N$ operation to return from the subclause verb *see* with the prn value 13 to the head noun *dog* in the main clause with the prn value 14:

### 3.3.15 Navigating with $V\uparrow N$ from *see* back to *dog* (arc 5)

The operations $N\downarrow V$ and $V\uparrow N$ handle the extrapropositional transition from a main clause to a adnominal subclause and back. They differ from $V\downarrow N$ (3.2.17, 4.1.16) and $N\uparrow V$ (3.2.18, 4.1.17), which handle the intrapropositional connection between the predicate and an adverbial prepositional phrase.

---

12 The operations $N\downarrow V$ and $V\uparrow N$ handle the extrapropositional transition from a main clause to a adnominal subclause and back. They differ from $V\downarrow N$ (3.2.17, 4.1.16) and $N\uparrow V$ (3.2.18, 4.1.17), which handle the intrapropositional connection between the predicate and an adverbial prepositional phrase.
3.3.16 **Navigating with N/V from dog back to bark (arc 6)**

<table>
<thead>
<tr>
<th>pattern level</th>
<th>level</th>
<th>⇒</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: (\beta)</td>
<td>sur: lexverb((\hat{\alpha}))</td>
<td>verb: (\alpha)</td>
<td>sur: barked_(_)</td>
</tr>
<tr>
<td>fnc: (\alpha)</td>
<td>arg: #(\beta) (Y)</td>
<td>prn: K</td>
<td>noun: dog</td>
</tr>
<tr>
<td>mdr: (Z)</td>
<td>prn: K</td>
<td>cat: snp</td>
<td></td>
</tr>
</tbody>
</table>

\(Z\) is \#-marked or NIL

Lexverb uses the core value bark, the sem value past, the \#-marking of the initial arg value, and the cat value decl (arative) of the goal proplet to realize the surface of the top verb together with the punctuation sign.

We conclude the section with point 7 on the to-do list [1.5.1]. Extending DBS.8 to a clausal adnominal modifier with a subject gap required the definition of the hear mode operations \(\mathcal{CN} \times \text{sub}_{\text{who}}\) (\(h9\)) and \(\text{sub}_{\text{who}} \cup \text{PRD}\) (\(h20\)), and of the speak mode operations \(\mathcal{N} \downarrow \mathcal{V}\) (\(s15\)) and \(\mathcal{V} \uparrow \mathcal{N}\) (\(s16\)). The hear and the speak mode derivation used six operation applications each.

3.3.17 **Comparing hear and speak mode surface segmentation**

**hear mode:**

\[1 2 3 4 5 6\]

The \(\cup\) dog \(\times\) which \(\cup\) saw \(\times\) Mary \(\times\) barked \(\cup\).

**speak mode:**

\[1 2 3 4 5 6\]

\(\mathcal{V}/\mathcal{N}\) The Dog \(\mathcal{N} \downarrow \mathcal{V}\) which saw \(\mathcal{V} \downarrow \mathcal{N}\) Mary \(\mathcal{N} \mathcal{V}\) \(\mathcal{V} \uparrow \mathcal{N}\) \(\mathcal{N} \mathcal{V}\) barked_.

The hear mode derivation runs straight through without a suspension. The speak mode derivation has two empty traversals.

3.4 **DBS.10: Clausal Adnominal Modifier with Object Gap**

The grammatical counterpart [13] to the example of the preceding section is a clausal adnominal with an object gap, e.g. The dog which Mary saw barked.

Let us follow standard procedure and begin with point 1 on the to-do list [1.5.1] by presenting the example and its content:
3.4.1 CONTENT OF THE SURFACE  The dog which Mary saw barked.

As contents, the only difference between this object gap construction and the subject gap construction 3.3.1 is in the respective arg feature of see: it is [arg: ∅ (person y)] in the subject gap content, but [arg: (person y) ∅] in the object gap content. In the English surface, this difference is coded by the word order, e.g. dog which [saw Mary] vs. dog which [Mary saw]. The surfaces of the two constructions may also differ in rudimentary case markings, as in man who [saw Mary] vs. man who(m) [Mary saw].

Let us continue with point 2 on the to-do list 1.5.1, i.e. the time-linear hear mode derivation in canonical DBS graph format, followed by point 3, i.e. the sequence of explicit hear mode operation applications which derive 3.4.1.

3.4.2 GRAPHICAL HEAR MODE DERIVATION OF THE CONTENT 3.4.1

For an overview of the grammatically possible and impossible constellations involving the combination of adnominal clauses and coordination, depending on the gap position, see NLC Sect. 9.5.
### Absorption with Simultaneous Substitution

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: v_1</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Result**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: (person y) 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Suspension**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: (person y)</th>
<th>Cat:</th>
<th>Sem: def sg - mf</th>
<th>Fnc:</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Cross-copying**

<table>
<thead>
<tr>
<th>Sur: Mary</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc:</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Absorption**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc: bark</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: bark</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: (person y) 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Absorption with Simultaneous Substitution**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc:</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: (person y) 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Suspension**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: (person y)</th>
<th>Cat:</th>
<th>Sem: def sg - mf</th>
<th>Fnc:</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Cross-copying**

<table>
<thead>
<tr>
<th>Sur: Mary</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc:</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Result**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc: bark</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: (person y) 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Suspension**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: (person y)</th>
<th>Cat:</th>
<th>Sem: def sg - mf</th>
<th>Fnc:</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Cross-copying**

<table>
<thead>
<tr>
<th>Sur: Mary</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc:</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Result**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc: bark</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: (person y) 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Absorption**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc: bark</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: (person y) 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Result**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc: bark</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: (person y) 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>

**Absorption**

<table>
<thead>
<tr>
<th>Sur:</th>
<th>Noun: dog</th>
<th>Cat: snp</th>
<th>Sem: def sg - mf</th>
<th>Fnc: bark</th>
<th>Mdr: (see 15)</th>
<th>Prn: 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: see</td>
<td>Cat:</td>
<td>Sem: which</td>
<td>Arg: (person y) 0</td>
<td>Mdd: (dog 14)</td>
<td>Prn: 15</td>
<td></td>
</tr>
</tbody>
</table>
The analysis uses the same lexical analysis of the subordinating conjunction \( v_I \) (which, NLC A.3.6) as [3.3.2]. In contradistinction to morphologically case-marked who vs. whom, which leaves it open whether \( \emptyset \) in its arg slot is in first (subject) or in second (object) position. In this way, assigning two lexical analyses to the same surface is avoided (surface compositionality).

The sequence of explicit hear mode operations begins with \( \text{DET}_U \text{CN} \):

### 3.4.3 ABSORBING dog INTO The WITH \( \text{DET}_U \text{CN} \) (line 1)

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[\text{noun: } N_{-R}]</td>
<td>[\text{noun: } n_{-n}]</td>
</tr>
<tr>
<td>[\text{cat: } X \text{ CN}]</td>
<td>[\text{cat: } \text{np}]</td>
</tr>
<tr>
<td>[\text{sem: } Y]</td>
<td>[\text{sem: } m_{-d}]</td>
</tr>
<tr>
<td>[\text{prn: } K]</td>
<td>[\text{prn: } 14]</td>
</tr>
</tbody>
</table>

Agreement condition as in [3.2.6]:

| sur: | dog |
| arg: | np |
| cat: | sn |
| sem: | sg |
| mdr: | mdr |
| nc: | nc |
| pc: | pc |
| prn: | prn: 14 |

Because which is not case marked morphologically, in contradistinction to who vs. whom, two operations apply next, \( \text{CN} \times \text{sub}^{\text{who}} \) and \( \text{CN} \times \text{sub}^{\text{whom}} \) (ambiguity). Having already shown \( \text{CN} \times \text{sub}^{\text{who}} \) in [3.3.4] and in anticipation of the arrival of a noun rather than a verb in [3.4.5], presenting the application of new \( \text{CN} \times \text{sub}^{\text{whom}} \) is sufficient for present purposes:

### 3.4.4 CROSS-COPYING dog AND which WITH \( \text{CN} \times \text{sub}^{\text{whom}} \) (line 2)

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[\text{noun: } \alpha]</td>
<td>[\text{noun: } \alpha]</td>
</tr>
<tr>
<td>[\text{cat: } \text{NP}]</td>
<td>[\text{cat: } \text{n}_{-p}]</td>
</tr>
<tr>
<td>[\text{mdr: } K]</td>
<td>[\text{arg: } \emptyset]</td>
</tr>
<tr>
<td>[\text{mdd: } (\text{V}_R K+1)]</td>
<td>[\text{mdd: } (\alpha K)]</td>
</tr>
</tbody>
</table>

\( \beta \in \{\text{who(m), which}\}; \text{NP} \in \{\text{np, npn, obq}\}; \text{NP}' \in \{d', a', be', hv', do'\} \)

| sur: | dog |
| arg: | np |
| cat: | sn |
| sem: | sg |
| mdr: | mdd |
| nc: | nc |
| pc: | pc |
| prn: | prn: 14 |

| sur: | which |
| arg: | np |
| cat: | sn |
| sem: | def sg |
| mdr: | mdd |
| nc: | nc |
| pc: | pc |
| prn: | prn: 14 |

| sur: | dog |
| arg: | np |
| cat: | sn |
| sem: | def sg |
| mdr: | mdd |
| nc: | nc |
| pc: | pc |
| prn: | prn: 14 |

| sur: | verb: \( v_1 \) |
| arg: | np |
| cat: | \( \text{V}_R \) |
| sem: | \( \alpha \) |
| mdr: | (V,R K+1) |
| mdd: | (\alpha K) |

| sur: | verb: \( v_1 \) |
| arg: | np |
| cat: | \( \text{V}_R \) |
| sem: | \( \alpha \) |
| mdr: | (V,R K+1) |
| mdd: | (\alpha K) |
In the input, the grammatical role of which is neutral, as shown by the middle position of ∅ in the arg slot of v_I. In the output, which is interpreted as the object gap by moving ∅ to the second (object) arg position. As in 3.3.4, the cat slot of the second output proplet specifies the range of permissible valency positions, though here for objects rather than subjects.

In the following derivation step, v_I and mary cannot be combined because of the missing information regarding agreement and valency positions in v_I. Therefore, new sub^{whom~SBJ} is a suspension:

### 3.4.5 Suspending v_I and Mary with sub^{whom~SBJ} (line 3)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: V_n</td>
<td>sur: Mary</td>
</tr>
<tr>
<td>sem: α</td>
<td>noun: (person y)</td>
</tr>
<tr>
<td>prn: K</td>
<td>cat: snp</td>
</tr>
<tr>
<td></td>
<td>sem: nm m</td>
</tr>
<tr>
<td></td>
<td>mdd: (dog 14)</td>
</tr>
<tr>
<td></td>
<td>nc: pc: prn: 15</td>
</tr>
<tr>
<td>α ∈ {whom, which}</td>
<td>sur: mary</td>
</tr>
<tr>
<td></td>
<td>noun: (person y)</td>
</tr>
<tr>
<td></td>
<td>cat: snp</td>
</tr>
<tr>
<td></td>
<td>sem: nm m</td>
</tr>
<tr>
<td></td>
<td>arg: ∅</td>
</tr>
<tr>
<td></td>
<td>mdd: (dog 14)</td>
</tr>
<tr>
<td></td>
<td>nc: pc: prn: 15</td>
</tr>
</tbody>
</table>

With the arrival of the predicate, standard SBJ×PRD applies:

### 3.4.6 Cross-copying Mary and saw with SBJ×PRD (line 4a)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: α</td>
<td>sur: saw</td>
</tr>
<tr>
<td>cat: NP</td>
<td>verb: see</td>
</tr>
<tr>
<td>fnc: β</td>
<td>cat: n’a v</td>
</tr>
<tr>
<td>prn: K</td>
<td>sem: past</td>
</tr>
<tr>
<td></td>
<td>arg: ∅</td>
</tr>
<tr>
<td></td>
<td>mdd: (dog 14)</td>
</tr>
<tr>
<td></td>
<td>nc: pc: prn: 15</td>
</tr>
</tbody>
</table>

The arrival and completion of the predicate allows to compensate the suspension between the conjunction sub^{whom} and the subclause by absorbing the predicate of the subclause into the conjunction with new sub^{whom~UPRD}:
3.4.7 ABSORBING see INTO v_1 WITH sub\textsuperscript{whom} \textsubscript{PRD} (line 4b)

The operation combines the respective arg and sem values of the two input proplets in the single output proplet and automatically replaces all occurrences of v_1 in any proplet at the current now front with see by simultaneous substitution. The operation also cancels by #-marking the object valency position in the cat slot of the resulting see proplet.

Having completed the time-linear hear mode processing of the subclause, it remains to complete the main clause with standard SBJ×PRD:

3.4.8 CROSS-COPYING dog AND barked WITH SBJ×PRD (line 5)

The presence of a subclause is indicated by the feature [mdr: (see 15)] in the dog proplet, introduced by CN×sub\textsuperscript{whom} in 3.4.4.

\textsuperscript{14} Without the who/whom alternative still present in English, we could have postponed the case assignment to line 4b (3.4.7), thus avoiding the ambiguity.
Finally, the period is absorbed into the top predicate by $S \cup IP$:

3.4.9 **ABSORBING $\cdot$ INTO bark WITH $S \cup IP$** (line 6)

Following standard procedure, we continue with point 4 on the to-do list 1.5.1, i.e. the DBS graph structure of the speak mode, based on the content derived in 3.4.2.

3.4.10 **GRAPH ANALYSIS UNDERLYING PRODUCTION OF** 3.4.1

(i) *SRG (semantic relations graph)*

(ii) *signature*

(iii) *NAG (numbered arcs graph)*

(iv) *surface realization*
Compared to 3.3.9 (subject gap), the object gap is indicated graphically by the missing \-branch under see.

The next item on the to-do list 1.5.1 is point 5, i.e. the list of speak mode operation names with surface realizations:

### 3.4.11 SEQUENCE OF OPERATION NAMES AND SURFACE REALIZATIONS

<table>
<thead>
<tr>
<th>Arc</th>
<th>Operation</th>
<th>Surface Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V\N</td>
<td>from bark to dog</td>
</tr>
<tr>
<td>2</td>
<td>N↓V</td>
<td>from dog to see</td>
</tr>
<tr>
<td>3</td>
<td>V/N</td>
<td>from see to mary</td>
</tr>
<tr>
<td>4</td>
<td>N\V</td>
<td>from mary to see</td>
</tr>
<tr>
<td>5</td>
<td>V↑N</td>
<td>from see to dog</td>
</tr>
<tr>
<td>6</td>
<td>N\V</td>
<td>from dog to bark</td>
</tr>
</tbody>
</table>

The speak mode operation sequences for clausal adnominals with a subject gap 3.3.10 and an object gap 3.4.11 differ in arc 3, which is V\N in 3.3.13 but V/N in 3.4.14 and in arc 4, which is N\V in 3.3.14 but N/V in 3.4.15.

The navigation is started by an activation of the top predicate. From there it continues autonomously along the semantic relations between proplets in the agent’s memory.

We continue with the sequence of explicit speak mode operations (1.5.1 6):

### 3.4.12 NAVIGATING WITH V/N FROM bark to dog (arc 1)

\[ V/N \begin{array}{c}
\text{ Pattern Level} \\
\begin{array}{c}
\text{verb: } \alpha \\
\text{arg: } \beta X \\
\text{prn: } K
\end{array}
\end{array} \Rightarrow \begin{array}{c}
\text{Content Level} \\
\begin{array}{c}
\text{sur: lexnoun(} \hat{\beta} \text{)} \\
\text{noum: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K
\end{array}
\end{array}
\]

#-mark \( \beta \) in the arg slot of proplet \( \alpha \)

The goal proplet offers two possible continuations, [fnc: bark] and [mdr: (see 15)]. As determined by the hear mode, the latter must be applied first:

\[ \text{To enforce realization of the original input surface, the choice between alternative speak mode continuations may be guided by continuation values which have been marked procedurally (invisibly) during the hear mode derivation. Alternatively, all possible speak mode variants for a given content (paraphrases) may be derived systematically.} \]

\[ ^{15} \text{To enforce realization of the original input surface, the choice between alternative speak mode continuations may be guided by continuation values which have been marked procedurally (invisibly) during the hear mode derivation. Alternatively, all possible speak mode variants for a given content (paraphrases) may be derived systematically.} \]
3. Optional Functor-Arguments and Coordination

3.4.13 NAVIGATING WITH N↓V FROM dog TO see (arc 2)

\[
\begin{align*}
\text{pattern level} & : \quad \text{noun: } \alpha \\
& \quad \text{mdr: } (\beta K+1) \\
& \quad \text{prn: } K \\
\uparrow & \\
\text{content level} & : \quad \text{sur: } \text{lexverb}(\beta) \\
& \quad \text{verb: } \beta \\
& \quad \text{mdd: } (\alpha K) \\
& \quad \text{prn: } K+1
\end{align*}
\]

\#-mark \(\alpha\) in the mdd slot of proplet \(\beta\)

The next two traversals complete the subclause:

3.4.14 NAVIGATING WITH V/N FROM see TO (person y) (arc 3)

\[
\begin{align*}
\text{pattern level} & : \quad \text{verb: } \alpha \\
& \quad \text{arg: } \beta X \\
& \quad \text{prn: } K \\
\uparrow & \\
\text{content level} & : \quad \text{sur: } \text{see} \\
& \quad \text{cat: } \#a' v \\
& \quad \text{sem: } \text{which past} \\
& \quad \text{arg: } (\text{person y}) \emptyset \\
& \quad \text{mdd: } (\text{dog 14}) \\
& \quad \text{nc: } \\
& \quad \text{pc: } \\
& \quad \text{prn: } 15
\end{align*}
\]

\#-mark \(\beta\) in the arg slot of proplet \(\alpha\)

3.4.15 NAVIGATING WITH N/V FROM Mary BACK TO see (arc 4)

\[
\begin{align*}
\text{pattern level} & : \quad \text{noun: } \beta \\
& \quad \text{mdr: } Z \\
& \quad \text{prn: } K \\
\uparrow & \\
\text{content level} & : \quad \text{sur: } \text{Mary} \\
& \quad \text{noun: } (\text{person y}) \\
& \quad \text{cat: } \text{snp} \\
& \quad \text{sem: nm f} \\
& \quad \text{fnc: } \text{see} \\
& \quad \text{mdr: } \\
& \quad \text{nc: } \\
& \quad \text{pc: } \\
& \quad \text{prn: } 15
\end{align*}
\]

\#-mark \(\alpha\) in the fnc slot of proplet \(\beta\)

\(Z\) is NIL, or elementary and #-marked
3.4.16 Navigating with V↑N from see back to dog (arc 5)

\[
\begin{align*}
V↑N & \quad \text{(16)} \\
\begin{array}{l}
\text{pattern} \\
\text{level}
\end{array} & \quad \text{content} \\
\begin{array}{l}
\text{verb: } \beta \\
\text{mdd: } \#(\alpha \ K) \\
\text{prn: } K+1
\end{array} & \quad \begin{array}{l}
\text{sur: } \uparrow \\
\text{verb: see} \\
\text{cat: } \#n \ #a' \ v \\
\text{sem: which past} \\
\text{arg: } \#(\text{person } y) \ \emptyset \\
\text{mdr: } \#(\text{dog } 14) \\
\text{nc:} \\
\text{pc:} \\
\text{prn: } 15
\end{array} \\
\begin{array}{l}
\text{noun: } \alpha \\
\text{mdd: } \#(\beta \ K+1) \\
\text{prn: } K
\end{array} & \quad \begin{array}{l}
\text{sur: } \downarrow \\
\text{noun: dog} \\
\text{cat: snp} \\
\text{sem: def sg} \\
\text{fnc: bark} \\
\text{mdr: } \#(\text{see } 15) \\
\text{nc:} \\
\text{pc:} \\
\text{prn: } 14
\end{array}
\end{align*}
\]

3.4.17 Navigating with N/V from dog back to bark (arc 6)

\[
\begin{align*}
N/V & \quad \text{(2)} \\
\begin{array}{l}
\text{pattern} \\
\text{level}
\end{array} & \quad \text{content} \\
\begin{array}{l}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K
\end{array} & \quad \begin{array}{l}
\text{sur: } \uparrow \\
\text{noun: dog} \\
\text{cat: snp} \\
\text{sem: def sg} \\
\text{fnc: bark} \\
\text{mdr: } \#(\text{see } 15) \\
\text{nc:} \\
\text{pc:} \\
\text{prn: } 14
\end{array} \\
\begin{array}{l}
\Rightarrow
\end{array} & \quad \begin{array}{l}
\text{verb: } \alpha \\
\text{arg: } \#(\beta \ X) \\
\text{prn: } K
\end{array} \\
\begin{array}{l}
\Rightarrow
\end{array} & \quad \begin{array}{l}
\text{sur: } \downarrow \\
\text{barked_} \\
\text{verb: bark} \\
\text{cat: } \#n3' \ \text{decl} \\
\text{sem: past} \\
\text{arg: } \#\text{dog} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: } 14
\end{array}
\end{align*}
\]

Extending DBS.9 to an example class with a clausal adnominal modifier having an object gap required the definition of the two hear mode operations \(\text{sub}^\text{whom} \sim \text{SBJ} (h33)\) and \(\text{sub}^\text{whom} \cup \text{PRD} (h21)\). Counting the suspension in line 3 and its compensation in line 4b, the hear mode derivation used seven applications, and the speak mode derivation used six:

3.4.18 Comparing hear and speak mode surface segmentation

**hear mode:**

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4a & 4b & 5 & 6 \\
\text{The } \cup \text{ dog } \times \text{ which } \sim \text{ Mary } \times \cup \text{ saw } \times \text{ barked } \cup .
\end{array}
\]

**speak mode:**

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
\text{V/N The } \text{dog } \text{N↓V which } \text{V/N Mary N/V saw V↑N N/V barked}_. 
\end{array}
\]
There is a suspension in the hear mode derivation and the speak mode derivation has two empty traversals.

### 3.5 DBS.11: Clausal Adverbial Modification

Clausal adverbial modification differs from the clausal adnominal modification presented in Sects. 3.3 (subject gap) and 3.4 (object gap) in that it has (i) no gap and (ii) a comparatively free word order (NLC Sect. 7.5).

#### 3.5.1 CONTENT OF THE SURFACE

*When Fido barked Mary laughed.*

<table>
<thead>
<tr>
<th>sur:</th>
<th>[sur: fido]</th>
<th>[sur: mary]</th>
<th>[sur: laugh]</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: bark</td>
<td>noun: (dog x)</td>
<td>noun: (person y)</td>
<td>verb: laugh</td>
</tr>
<tr>
<td>cat: #n v</td>
<td>cat: snp</td>
<td>cat: snp</td>
<td>cat: #n' decl</td>
</tr>
<tr>
<td>sem: when</td>
<td>sem: nm m</td>
<td>sem: nm f</td>
<td>sem: past</td>
</tr>
<tr>
<td>arg: (dog x)</td>
<td>fnc: bark</td>
<td>fnc: laugh</td>
<td>arg: (person y)</td>
</tr>
<tr>
<td>mdd: (laugh 17)</td>
<td>mdr:</td>
<td>mdr: (bark 16)</td>
<td></td>
</tr>
<tr>
<td>nc:</td>
<td>pc:</td>
<td>pc:</td>
<td></td>
</tr>
<tr>
<td>pc: prn: 16</td>
<td>prn: 16</td>
<td>prn: 17</td>
<td>prn: 17</td>
</tr>
</tbody>
</table>

Consider the hear mode derivation in canonical DBS graph format:

#### 3.5.2 GRAPHICAL HEAR MODE DERIVATION OF THE CONTENT

*When Fido barked Mary laughed.*

[Diagram of the hear mode derivation with syntactic-semantic parsing and suspension and cross-copying annotations]
Similar to a subject clause (Sect. 2.5) and an object clause (Sect. 2.6), the subordinating conjunction of an adverbial clause is followed by the subclause subject. As in the other subclause constructions, there is no semantic relation between the conjunction and the subject in an adverbial clause. Therefore, the first step of the hear mode derivation in the present example is a suspension with simultaneous substitution.

---

**Table:**

<table>
<thead>
<tr>
<th>Step</th>
<th>Sentence</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>2b</td>
<td>When Mary fido barked, Fido barked at Mary.</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>3</td>
<td>When Fido barked, Fido barked.</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>4a</td>
<td>When Fido barked, Fido barked.</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>4b</td>
<td>When Fido barked, Fido barked.</td>
<td>![Diagram]</td>
</tr>
<tr>
<td>5</td>
<td>Result When Fido barked, Fido barked.</td>
<td>![Diagram]</td>
</tr>
</tbody>
</table>

---

16 Instead, there is a cross-copying between the subject and the predicate (line 2a) and an absorption of the predicate into the conjunction (line 2b). In order to ensure correct agreement, the absorption must follow the cross-copying.
3. Optional Functor-Arguments and Coordination

3.5.3 Suspending \( v_1 \) and Fido with \( \text{sub}^{\text{clause}} \sim \text{SBJ} \) (line 1)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{verb: } V_n )</td>
<td>( \text{verb: } V_n )</td>
</tr>
<tr>
<td>( \text{sem: } \alpha )</td>
<td>( \text{sem: } \alpha )</td>
</tr>
<tr>
<td>( \text{prn: } K )</td>
<td>( \text{prn: } K )</td>
</tr>
<tr>
<td>( \alpha \in {\text{that, when, where, how, why}} )</td>
<td>( \alpha \in {\text{that, when, where, how, why}} )</td>
</tr>
</tbody>
</table>

Using the same operation for a subject, object, and adverbiacl clause is possible not only because of pattern structures matching the input in principle, but also because of the comparatively liberal variable restriction on \( \alpha \).

The suspension between the subordinating conjunction and the subject is resolved by two operation applications in the next derivati on step. The first in line 2a connects Fido and barked with \( \text{SBJ} \times \text{PRD} \) by cross-copying their core values into the respective \( \text{fnc} \) and \( \text{arg} \) slots.

3.5.4 Cross-copying \( \text{fido} \) and barked with \( \text{SBJ} \times \text{PRD} \) (line 2a)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{noun: } \alpha )</td>
<td>( \text{verb: } \beta )</td>
</tr>
<tr>
<td>( \text{cat: } \text{NP} )</td>
<td>( \text{cat: } \text{NP} \times \text{X} \text{v} )</td>
</tr>
<tr>
<td>( \text{fnc: } )</td>
<td>( \text{arg: } \text{NP} \times \text{X} \text{v} )</td>
</tr>
<tr>
<td>( \text{prn: } K )</td>
<td>( \text{prn: } K )</td>
</tr>
</tbody>
</table>

Agreement conditions as in \( \text{2.5.3} \)

The second operation in line 2b compensates the suspension with \( \text{sub}^{\text{clause}} \cup \text{PRD} \), which absorbs the predicate, just completed by \( \text{SBJ} \times \text{PRD} \), into the subordinating conjunction:
3.5.5 Absorbing *bark* into \(v_1\) with \(\text{sub}^{\text{clause}\cup\text{PRD}}\) (line 2b)

At this point, the time-linear hear mode derivation has interpreted the input up to *When Fido barked*.

The next proplet provided by automatic word form recognition is the main clause subject *mary*. It is attached to the adverbial subclause by suspension:

3.5.6 Suspending *bark* and *Mary* with \(\text{sub}^{\text{when}\sim\text{NOM}}\) (line 3)

This operation is limited to assigning a \(\text{prn}\) value to the second input proplet, albeit an incremented one. Its variable restriction requires the first input proplet to represent an adverbial subclause and the next word to be a noun which can serve as a nominative. As usual, the subordinating conjunction is preserved in the first \(\text{sem}\) slot, here of *bark*.

The next word proplet *laugh* activates two hear mode operations, (i) standard \(\text{SBJ}\times\text{PRD}\) for continuing (and completing) the main clause (line 4a) and
(ii) $\text{sub}^\text{when} \times \text{PRD}$ for establishing the modifier-modified relation between the adverbial modifier clause and the higher predicate by cross-copying between bark and laugh (line 4b).

The next word proplet laugh matches the second input pattern of SBJ $\times$ PRD. Its first input pattern finds mary at the now front and applies as follows:

### 3.5.7 Cross-copying mary and laughed with SBJ $\times$ PRD (line 4a)

The next word proplet laugh also matches second input pattern of the new operation sub$^\text{when} \times \text{PRD}$ Its first input pattern finds bark at the now front and applies as well:

### 3.5.8 Cross-copying bark and laughed with sub$^\text{when} \times \text{PRD}$ (line 4b)

At this point, the time-linear hear mode derivation has interpreted When Fido barked, Mary laughed. It remains to add the period proplet . supplied by automatic word form recognition:

---

17 Superficially, the operations $\text{sub}^\text{when} \times \text{PRD}$ and $\text{sub}^\text{that} \times \text{PRD}$ look similar, but differ in that the former cross-copies into mdd-mdr slots and the latter into fnc-arg slots.
3.5.9 Absorbing INTO laugh with SUIP (Line 5)

This completes the hear mode derivation of an example showing an adverbial clause in preverbal position (cf. NLC Sect. 7.5).

In accordance with point 4 of the DBS laboratory set-up 1.5.1, we turn next to the canonical DBS graph analysis of the speak mode:

3.5.10 Graph analysis underlying production of (3.5.1)

(i) SRG (semantic relations graph)  (iii) NAG (numbered arcs graph)

(ii) signature

(iv) surface realization

In the paraphrase Mary laughed when Fido barked, the depth first arc numbering is consecutive.

The next item on the to-list 1.5.1 is point 5, i.e. the list of speak mode operation names with surface realizations:
3.5.11 SEQUENCE OF OPERATION NAMES AND SURFACE REALIZATIONS

arc 3: \( V \downarrow V \) from laugh to bark  
\[ \text{When} \]

arc 4: \( V / N \) from bark to fido  
\[ \text{Fido} \]

arc 5: \( N \uparrow V \) from fido to bark  
\[ \text{barked} \]

arc 6: \( V \uparrow V \) from bark to laugh  

arc 1: \( V / N \) from laugh to mary  
\[ \text{Mary} \]

arc 2: \( N \uparrow V \) from mary to laugh  
\[ \text{laughed} \]

The first navigation step \( V \downarrow V \) resembles the first navigation step \( V / V \) of the subject clause example 2.5.10 in that it moves from the top verb to the verb representing an initial subclause. They differ in that \( V / N \) navigates to a clausal subject argument, while \( V \downarrow V \) navigates to a clausal adverbial modifier.

3.5.12 NAVIGATING WITH \( V \downarrow V \) FROM \textit{laugh} TO \textit{bark} (arc 3)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \quad \text{content level} \\
\verb: \alpha & \quad [\text{sur: lexverb} (\hat{\beta})] \\
\text{mdr: } X \beta Y & \quad [\text{verb: } \beta] \\
\text{prn: } K & \quad [\text{mdd: } \alpha] \\
\text{content level} & \quad \Rightarrow \quad \text{content level} \\
\uparrow & \quad \downarrow \\
\text{sur: } & \quad [\text{verb: bark}] \\
\text{verb: } laugh & \quad [\text{cat: } \#n' \text{ v}] \\
\text{sem: past} & \quad [\text{sem: when past}] \\
\text{arg: } (\text{person } y) & \quad [\text{arg: } (\text{dog } x)] \\
\text{mdd: } (\text{bark } 16) & \quad [\text{mdd: } (\text{laugh } 17)] \\
\text{nc: } & \quad [\text{nc: }] \\
\text{pc: } & \quad [\text{pc: }] \\
\text{prn: } 17 & \quad [\text{prn: } 16] \\
\text{sur: } & \quad [\text{sur: } Fido] \\
\text{noun: } & \quad [\text{noun: } (\text{dog } x)] \\
\text{cat: } & \quad [\text{cat: } \text{snp}] \\
\text{sem: } & \quad [\text{sem: } \text{nm m}] \\
\text{fnc: } & \quad [\text{fnc: bark}] \\
\text{mdr: } & \quad [\text{mdr: }] \\
\text{nc: } & \quad [\text{nc: }] \\
\text{pc: } & \quad [\text{pc: }] \\
\text{prn: } 16 & \quad [\text{prn: } 16] \\
\end{align*}
\]

The navigation continues from the lower predicate to the lower subject.

3.5.13 NAVIGATING WITH \( V / N \) FROM \textit{bark} TO \textit{fido} (arc 4)
3.5.14 **Navigating with N/V from *fido* back to *bark* (arc 5)**

\[
\begin{array}{c}
\text{pattern level} \\
\text{noun: } \beta \\
fnc: \alpha \\
mdr: Z \\
prn: K \\
\Rightarrow \\
\text{sur: lexverb(\(\hat{\alpha}\))} \\
\text{verb: } \alpha \\
\text{arg: } \#\beta X \\
\text{prn: K} \\
\end{array}
\]

Z is NIL, or elementary and \#-marked

\[
\begin{array}{c}
\text{content level} \\
\text{sur: *fido*} \\
noun: \text{(dog } x) \\
cat: \text{snp} \\
sem: \text{nm m} \\
fnc: \text{bark} \\
mdr: \\
nc: \\
pc: \\
prn: 16 \\
\end{array}
\]

\[
\begin{array}{c}
\text{sur: *barked*} \\
noun: \text{verb: } \alpha \\
cat: \text{#n' v} \\
sem: \text{when past} \\
arg: \#(dog } x) \\
mdr: \text{(laugh 17)} \\
nc: \\
pc: \\
prn: 16 \\
\end{array}
\]

3.5.15 **Navigating with V↑V from *bark* back to *laugh* (arc 6)**

\[
\begin{array}{c}
\text{pattern level} \\
\text{verb: } \alpha \\
mdr: X \#\beta Y \\
prn: K \\
\Rightarrow \\
\text{sur: lexverb(\(\hat{\beta}\))} \\
\text{verb: } \beta \\
\text{mdr: } \alpha \\
\text{prn: K+1} \\
\end{array}
\]

\[
\begin{array}{c}
\text{content level} \\
\text{sur: *bark*} \\
\text{cat: #n' v} \\
\text{sem: when past} \\
\text{arg: #(dog } x) \\
\text{mdr: #(laugh 17)} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: 16} \\
\end{array}
\]

\[
\begin{array}{c}
\text{sur: *laugh*} \\
\text{cat: #n' v} \\
\text{sem: past} \\
\text{arg: #(person } y) \\
\text{mdr: #(bark 16)} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: 17} \\
\end{array}
\]

3.5.16 **Navigating with V/N from *laugh* to *Mary* (arc 1)**

\[
\begin{array}{c}
\text{pattern level} \\
\text{verb: } \alpha \\
arg: \#\beta X \\
prn: K \\
\Rightarrow \\
\text{sur: lexnoun(\(\hat{\beta}\))} \\
noun: \beta \\
fnc: \alpha \\
prn: K \\
\end{array}
\]

\[
\begin{array}{c}
\text{content level} \\
\text{sur: *laugh*} \\
\text{cat: #n' v} \\
\text{sem: past} \\
\text{arg: #(person } y) \\
\text{mdr: #(bark 16)} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: 17} \\
\end{array}
\]

\[
\begin{array}{c}
\text{sur: *Mary*} \\
noun: \text{(person } y) \\
cat: \text{snp} \\
sem: \text{nm f} \\
fnc: \text{laugh} \\
mdr: \\
nc: \\
pc: \\
prn: 17 \\
\end{array}
\]
3.5.17 **Navigating with N/V from Mary back to laugh (arc 2)**

The surface realization resembles that in 3.3.16.

Extending DBS.10 to an example class with clausal adverbial modification required the definition of four additional operations, namely $\text{sub}^{\text{when} \sim \text{NOM}}$ (h36) and $\text{sub}^{\text{when} \times \text{PRD}}$ (h10) for the hear mode, and $\text{V}_\downarrow \text{V}$ (s17) and $\text{V}_\uparrow \text{V}$ (s18) for the speak mode. Counting the suspensions in lines 1 and 3 and their compensation in lines 2b and 4b, the hear mode derivation used seven operation applications, while the speak mode derivation used six:

3.5.18 **Comparing hear and speak mode surface segmentation**

**Hear mode:**

1 2a 2b 3 4a 4b 5

When $\sim$ Fido $\times$ $\cup$ barked $\times$ Mary $\times$ $\times$ laughed $\cup$ .

**Speak mode:**

3 4 5 6 1 2

$\text{V}_\downarrow \text{V}$ When $\text{V}_/\text{N}$ Fido $\text{N}/\text{V}$ barked $\text{V}_\uparrow \text{V}$ $\text{V}_/\text{N}$ Mary $\text{N}/\text{V}$ laughed_.

The numbering of the speak mode operations corresponds to the arcs traversed in 3.5.10. There are two suspensions in the hear mode derivation and an empty traversal in the speak mode derivation.

3.6 **DBS.12: Coordination**

The functor-argument relations of subject/predicate, object/predicate, adnominal/noun, and adverbial/verb, at the elementary, phrasal, and clausal levels of grammatical, complexity combine a “lower” proplet with a “higher” one
(hypotaxis). The conjunct–conjunct relation, in contrast, combines two proplets at the same level (parataxis), intra- and extrapropositionally.

Consider the following example of an intrapropositional coordination serving as a subject and its content as a set of proplets:

3.6.1 CONTENT OF SURFACE Fido, Tucker, and Buster snored loudly.

The connection between the conjuncts is coded by the nc (next conjunct) and pc (previous conjunct) values: the first conjunct points forward to the second conjunct via its nc value, the second conjunct points forward to the third via its nc value and backward to the first via its pc value, etc. The initial conjunct has an empty pc slot, just as the final conjunct has an empty nc slot.

A chain of conjuncts is integrated into the functor-argument structure by using only the initial conjunct with an &-marked prn value \(^{18}\) (NLC Sects. 8.2, 8.3). For example, the core value of the &-marked fido proplet appears in the arg slot of snore and the core value of snore appears in the fnc slot of the fido proplet. The core values of the non-initial conjunct proplets tucker and buster, in contrast, do not appear in the arg slot of snore and snore does not appear in the fnc slots of the non-initial tucker and buster proplets.

The content results from the following hear mode derivation:

3.6.2 GRAPHICAL HEAR MODE DERIVATION OF THE CONTENT

\(^{18}\) The treatment coordination in NLC # marks all instances of the initial conjunct’s core value. The present treatment, inspired by the upscaling to long-distance dependency (Sect. 5.5), # marks the prn value of the initial conjunct instead, which is simpler. The overall approach remains the same.
syntactic–semantic parsing

1. sur: Fido
cat: snp
sem: nm m
fnc: pc: prn: 18
noun: (dog x)
cat: snp
sem: nm m
fnc: pc: prn: 18

2. sur: Tucker
cat: snp
sem: nm m
fnc: pc: prn: 18
noun: (dog y)
cat: snp
sem: nm m
fnc: pc: prn: 18

3. sur: and
noun: and
sem: and
prn: 18

3a. sur: Tucker
noun: (dog y)
cat: snp
sem: nm m
fnc: pc: prn: 18

3b. sur: Buster
noun: (dog z)
cat: snp
sem: and nm m
fnc: pc: prn: 18

4. sur: snored
verb: snore
cat: snp
sem: past
fnc: prn: 18

5. sur: loudly
adj: loud
cat: adv
sem: pad
fnc: prn: 18

6. sur: v_2
verb: v_2
cat: adv
sem: v' decl
fnc: prn: 18
The subject of the construction is a noun coordination. The conjuncts consist of name proplets. The verb is one-place and is modified by a clause-final elementary adverb. The construction is intrapropositional.

Following standard procedure, we continue with point 3 on the to-do list 1.5.1, i.e. the complete sequence of explicit hear mode operation applications. It begins with the coordination of the conjuncts Fido and Tucker, provided by automatic word form recognition. The initial coordination operation is called \( \text{N}_1 \times \text{N} \). It cross-copies the core values into the respective \text{nc} and \text{pc} slots of the first two conjuncts and marks the \text{prn} value of the initial conjunct with \& (because only the initial conjunct will be connected to the functor-argument structure, NLC 8.1.4).

3.6.3 CROSS-COPYING Fido AND Tucker WITH \( \text{N}_1 \times \text{N} \) (line 1)

An initial noun conjunct will receive its \text{fnc} value when it is combined with the verb, here in line 4 3.6.7. Non-initial noun conjuncts, in contrast, will not receive a \text{fnc} value; if needed for retrieval, the \text{fnc} value must be obtained from the initial conjunct by navigating backwards along the \text{pc} connections.

If there are more than two noun conjuncts, the nonfinal ones are connected by a coordination operation called \( \text{N} \times \text{N} \), which resembles \( \text{N}_1 \times \text{N} \) except that no \&-marker is introduced. The coordination of verbs and adjectives is based on similar operations, but with different core attributes.
In the hear mode, the approaching end of a coordination is announced by a coordinating conjunction, such as and or or. Semantically, the conjunction should be coded into the final conjunct, which has not arrived yet. Therefore the next operation to apply is a suspension:

3.6.4 **SUSPENDING tucker AND and WITH N~CNJ** (line 2)

![Pattern level diagram]

\[
\text{N} \sim \text{CNJ (line 2)}
\]

As usual in a suspension, the operation is limited to assigning a prn value to the second output proplet.

The suspension is compensated by the operation CNJUN. It integrates the conjunction into the final conjunct, which arrives next:

3.6.5 **ABSORRING and INTO Buster with CNJUN** (line 3a)

![Pattern level diagram]

\[
\text{CNJUN (line 3a)}
\]

\[19\] The conjunction but not may initiate a second coordination, as in Fido, Tucker, and Buster, but not Bailey, Molly, and Daisy.
To allow a general application of the following operation \( N \times N \), regardless of whether or not the current output is the final conjunct, \( \text{CNJ} \cup N \) does not provide a \( \text{prn} \) value to the output proplet, similar to \( \text{CNJ} \cup V \) (5.2.10).

The coordination may be continued indefinitely with \( N \times N \), but only if the first input proplet does not have a conjunction in its \( \text{sem} \) slot (see condition):

### 3.6.6 CROSS-COPYING tucker AND buster WITH \( N \times N \) (line 3b)

\[
\begin{align*}
\text{pattern level} & \quad \text{content level} \\
\begin{array}{l}
\begin{array}{l}
\text{noun: } \alpha \\
\text{nc: } \gamma \\
\text{prn: } K \\
\end{array} & \\
\begin{array}{l}
\text{noun: } \beta \\
\text{nc: } \eta \\
\text{prn: } K \\
\end{array} & \\
\begin{array}{l}
\text{noun: } \alpha & \\
\text{nc: } \gamma & \\
\text{prn: } K & \\
\end{array} & \\
\begin{array}{l}
\text{noun: } \beta & \\
\text{nc: } \eta & \\
\text{prn: } K & \\
\end{array}
\end{array}
\Rightarrow
\begin{array}{l}
\text{sur: } \text{tucker} & \\
\text{noun: } \text{(dog } y) & \\
\text{cat: } \text{snp} & \\
\text{sem: } \text{nm m} & \\
\text{fnc: } & \\
\text{prn: } 18 & \\
\end{array}
\end{align*}
\]

The \( \text{sem} \) slot of \( \alpha \) may not contain a conjunction.

Next, automatic word form recognition provides the finite verb form \text{snored}, which activates standard \( \text{SBJ} \times \text{PRD} \) by matching its second input pattern:

### 3.6.7 CROSS-COPYING fido AND snored WITH \( \text{SBJ} \times \text{PRD} \) (line 4)

\[
\begin{align*}
\text{pattern level} & \quad \text{content level} \\
\begin{array}{l}
\begin{array}{l}
\text{noun: } \alpha \\
\text{cat: } \text{NP} & \\
\text{fnc: } & \\
\text{prn: } K & \\
\end{array} & \\
\begin{array}{l}
\text{verb: } \beta \\
\text{cat: } \text{NP} & \\
\text{arg: } & \\
\text{prn: } K & \\
\end{array} & \\
\begin{array}{l}
\text{noun: } \alpha & \\
\text{verb: } \text{snore} & \\
\text{cat: } \text{NP} & \\
\text{arg: } & \\
\text{prn: } K & \\
\end{array} & \\
\begin{array}{l}
\text{verb: } \beta & \\
\text{cat: } \text{NP} & \\
\text{arg: } & \\
\text{prn: } K & \\
\end{array}
\end{array}
\Rightarrow
\begin{array}{l}
\begin{array}{l}
\text{sur: } \text{fido} & \\
\text{noun: } \text{(dog } x) & \\
\text{cat: } \text{snp} & \\
\text{sem: } \text{nm m} & \\
\text{fnc: } & \\
\text{prn: } 18 & \\
\end{array} & \\
\begin{array}{l}
\text{verb: } \text{snore} & \\
\text{cat: } \text{snp} & \\
\text{sem: } \text{nm m} & \\
\text{fnc: } \text{snore} & \\
\text{arg: } \text{(dog } y) & \\
\text{pc: } & \\
\text{prn: } 18 & \\
\end{array} & \\
\begin{array}{l}
\text{sur: } \text{snored} & \\
\text{noun: } \text{(dog } x) & \\
\text{cat: } \text{snp} & \\
\text{sem: } \text{nm m} & \\
\text{fnc: } \text{snore} & \\
\text{arg: } \text{(dog } y) & \\
\text{pc: } & \\
\text{prn: } 18 & \\
\end{array} & \\
\begin{array}{l}
\text{sur: } \text{snored} & \\
\text{noun: } \text{(dog } x) & \\
\text{cat: } \text{snp} & \\
\text{sem: } \text{nm m} & \\
\text{fnc: } \text{snore} & \\
\text{arg: } \text{(dog } y) & \\
\text{pc: } & \\
\text{prn: } 18 & \\
\end{array}
\end{array}
\end{align*}
\]
If the subject is a noun coordination, the operation \textbf{SBJ} \times \textbf{PRD} cross-copies only the initial conjunct, recognizable by its empty \textit{pc} slot.\footnote{A non-coordinated subject and the initial conjunct of a coordinated subject have in common that their \textit{pc} slot is empty. The beginning of a grammatical object, in contrast, cancels a non-initial \textit{arg} slot of the predicate before the presence or absence of an object coordination is decided by the hear mode input (NLC 8.2.5).} Once the initial conjunct has canceled the NP' valency position, it cannot be filled again.\footnote{With predicate gapping as a special case (Sect. 5.3).}

The elementary adj \textit{loudly} provides a comparison between a coordinated argument and an adverbial modification. It is added with \textbf{FV} \times \textbf{ADV} (3.1.6):

3.6.8 \textbf{CROSS-COPYING snore AND loudly WITH FV} \times \textbf{ADV} (line 5)

![Pattern and content level diagram]

The continuation features of \textit{snore} for the speak mode are [\textit{arg}: (dog x)] and [\textit{mdr}: loud].

The hear mode derivation\footnote{3.6.2} concludes with adding the period.

3.6.9 \textbf{ABSORBING \bullet INTO snore WITH SUIP} (line 6)

![Pattern and content level diagram]

Agreement conditions as in 2.1.3.
The verb proplet resulting from the hear mode derivation has the single subject value (dog x).

Let us follow standard procedure and turn next to the associated speak mode derivation. In accordance with the DBS laboratory set-up, this derivation must map the content 3.6.1 back into the original surface. We begin with point 4 of the to-do list 1.5.1 i.e. the canonical DBS graph analysis showing the semantic relations between proplets and the navigation along these relations (selective activation): 3.6.10 Graph analysis underlying production of 3.6.1

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

\[
\begin{align*}
\text{snore} & \quad \text{loud} \\
\text{fido} & \rightarrow \text{tucker} \rightarrow \text{buster} \\
\text{V} & \quad \text{A} \\
\end{align*}
\]

(iv) surface realization

\[
\begin{align*}
\text{Fido} & \quad \text{Tucker} & \quad \text{and_Buster} & \quad \text{snored} & \quad \text{loudly} \\
\text{V/N} & \quad \text{N→N} & \quad \text{N→N} & \quad \text{N→N} & \quad \text{N→N} & \quad \text{V/A} & \quad \text{A/V} \\
\end{align*}
\]

The direction of N→N traversals is indicated by the associated arc numbers.

We continue with point 5 on the to-do list 1.5.1 i.e. the list of speak mode operation names with surface realizations:

3.6.11 Sequence of operation names and surface realizations

\[
\begin{align*}
\text{arc 1: V/N from snore to fido} & \quad \text{Fido} \quad 3.6.12 \\
\text{arc 2: N→N from fido to tucker} & \quad \text{Tucker} \quad 3.6.13 \\
\text{arc 3: N→N from tucker to buster} & \quad \text{and_Buster} \quad 3.6.14 \\
\text{arc 4: N←N from buster to tucker} & \quad 3.6.15 \\
\text{arc 5: N←N from tucker to fido} & \quad 3.6.16 \\
\text{arc 6: N/V from fido to snore} & \quad \text{snored} \quad 3.6.17 \\
\text{arc 7: V↓A from snore to loud} & \quad \text{loudly} \quad 3.6.18 \\
\text{arc 8: A↑V from loud to snore} & \quad . \quad 3.6.19 \\
\end{align*}
\]

The forward navigation uses N→N while the backward navigation uses N←N

The next point on the to-do list 1.5.1 is the complete sequence of explicit speak mode operation applications. It begins with the traversal of arc 1 with standard V/N (cf. 1.6.5 4):
3.6.12 Navigating with $V_f N$ from *snore to fido* (arc 1)

\[
\begin{align*}
\text{pattern level} & : \\
\begin{array}{c}
\text{content level} \\
\text{sur: snore} \\
\text{cat: \#n' decl +ip} \\
\text{sem: past} \\
\text{arg: (dog x)} \\
\text{mdr: loud} \\
\text{prn: 18} \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
\text{pattern level} & : \\
\begin{array}{c}
\text{content level} \\
\text{sur: Fido} \\
\text{noun: (dog x)} \\
\text{cat: snp} \\
\text{sem: nm m} \\
\text{fnc: snore} \\
\text{nc: (dog y)} \\
\text{pc: (dog x)} \\
\text{prn: 18}\& \\
\end{array}
\end{align*}
\]

Because the goal proplet is a name, lexnoun realizes *Fido* from the corresponding marker in the sur slot (and not the core value).22

Using the continuation feature [nc: (dog y)] of the goal proplet, the navigation enters the coordination and traverses arc 2 with the operation N→N:

3.6.13 Navigating with N→N from *fido to tucker* (arc 2)

\[
\begin{align*}
\text{pattern level} & : \\
\begin{array}{c}
\text{content level} \\
\text{sur: fido} \\
\text{noun: (dog x)} \\
\text{cat: snp} \\
\text{sem: nm m} \\
\text{fnc: nc: (dog y)} \\
\text{pc: pc: (dog x)} \\
\text{prn: 18}\& \\
\end{array}
\end{align*}
\]

\[
\begin{align*}
\text{pattern level} & : \\
\begin{array}{c}
\text{content level} \\
\text{sur: Tucker} \\
\text{noun: (dog y)} \\
\text{cat: snp} \\
\text{sem: nm m} \\
\text{fnc: nc: (dog z)} \\
\text{pc: pc: (dog y)} \\
\text{prn: 18} \\
\end{array}
\end{align*}
\]

The sur slot of the input proplet shows a name marker which is written in default font and lower case. It was used for surface realization in 3.6.12. The sur value of the output proplet shows the result of the current lexnoun application, *Tucker*. Written in Helvetica and a capitalized initial letter, it temporarily covers the name marker until the name surface is moved to the agent’s action component for word form realization and the name marker shows up again.

The next navigation step uses the continuation feature [nc: (dog z)] of the above goal proplet and traverses arc 3 with N→N:

\[\text{(For the matching between K and 18& see 1.6.5)} 4.\]
3.6.14 **Navigating with N→N from tucker to buster** (arc 3)

\[
\begin{align*}
\text{pattern level} & : \begin{cases}
\text{noun: } \alpha \\
\text{pc: } \beta \\
\text{prn: } K
\end{cases} \Rightarrow \begin{cases}
\text{sur: lexnoun(}\beta\text{)} \\
\text{noun: } \beta \\
\text{pc: } \alpha \\
\text{prn: } K
\end{cases} \\
\text{content level} & : \begin{cases}
\text{sur: } \text{tucker} \\
\text{noun: } (\text{dog y}) \\
\text{cat: } \text{snp} \\
\text{sem: } \text{nm m} \\
\text{fnc: } \\
\text{mdr: } \\
\text{nc: } (\text{dog z}) \\
\text{pc: } (\text{dog x}) \\
\text{prn: } 18
\end{cases} \\
\quad \uparrow \quad \downarrow & \begin{cases}
\text{sur: } \text{and_Buster} \\
\text{noun: } (\text{dog z}) \\
\text{cat: } \text{snp} \\
\text{sem: } \text{and nm m} \\
\text{fnc: } \\
\text{mdr: } \\
\text{nc: } (\text{dog z}) \\
\text{pc: } (\text{dog y}) \\
\text{prn: } 18
\end{cases}
\end{align*}
\]

As a name, the sur slot of the goal proplet contains the marker “buster.” Together with the initial sem value and, it is used by lexnoun to realize the surface and_Buster.

At this point, the navigation has traversed the proplets fido tucker buster via their nc values and the end of the coordination has been reached. This is shown by the goal proplet having neither an mdr nor an nc continuation value. To return to the top verb, the new speak mode operation N←N applies as follows:

3.6.15 **Navigating with N←N from buster back to tucker** (arc 4)

\[
\begin{align*}
\text{pattern level} & : \begin{cases}
\text{noun: } \alpha \\
\text{pc: } \beta \\
\text{prn: } K
\end{cases} \Rightarrow \begin{cases}
\text{noun: } \beta \\
\text{pc: } \alpha \\
\text{prn: } K
\end{cases} \\
\text{content level} & : \begin{cases}
\text{sur: } \text{buster} \\
\text{noun: } (\text{dog z}) \\
\text{cat: } \text{snp} \\
\text{sem: } \text{and nm m} \\
\text{fnc: } \\
\text{mdr: } \\
\text{nc: } (\text{dog z}) \\
\text{pc: } (\text{dog y}) \\
\text{prn: } 18
\end{cases} \\
\quad \uparrow \quad \downarrow & \begin{cases}
\text{sur: } \text{tucker} \\
\text{noun: } (\text{dog y}) \\
\text{cat: } \text{snp} \\
\text{sem: } \text{nm m} \\
\text{fnc: } \\
\text{mdr: } \\
\text{nc: } (\text{dog z}) \\
\text{pc: } (\text{dog x}) \\
\text{prn: } 18
\end{cases}
\end{align*}
\]

In contradistinction to N→N (3.6.13, 3.6.14), which uses the nc value of its start proplet to move forward, N←N uses the pc value of its start proplet to move backward.

To get back to the initial conjunct, N←N applies once more.

---

23 For more on intra- and extrapropositional coordination in DBS see CLaTR 3.2.4–3.2.6, 4.4.2, 7.5.9–7.5.12, 8.3.1, 9.1.4, 9.4.3, 9.4.5, and NLC 6.4.4, 8.1.1–8.1.4, 8.2.1–8.3.6, 9.1.1, 9.1.2, 9.2.1–9.6.9, 11.6.5, 12.3.4, 13.3.2, 13.3.5, 13.4.2, 13.4.5, 14.3.2, 14.4.2, 15.3.3.
3.6.16 **Navigating with \( N \leftarrow N \) from tucker back to fido (arc 5)**

As the initial conjunct, the proplet fido provides the continuation value snore, enabling the operation \( N \nearrow V \) to return to the predicate:

3.6.17 **Navigating with \( N \nearrow V \) from fido back to snore (arc 6)**

The \#-marking of the initial arg value resulted from the instruction in \[3.6.12\]. It remains to traverse and realize the adverbial loudly with the operation \( V \downarrow A \) and to return to the verb to realize the period with the operation \( A \uparrow V \). The operations have been introduced in \[3.1.13\] and \[3.1.14\] with an example in which an elementary adverb is in sentence-initial position; here, in contrast, the elementary adverb is sentence-final.
3.6.18 **Navigating with \( \downarrow A \)** from *snore* to *loud* (arc 7)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \quad \text{content level} \\
\begin{array}{l}
\text{verb: } \alpha \\
\text{mdr: } \#X \beta Y \\
\text{prn: } K
\end{array}
& \quad \begin{array}{l}
\text{sur: } \text{lexadj}(\beta) \\
\text{adj: } \beta \\
\text{mdr: } \alpha \\
\text{prn: } K
\end{array}
\end{align*}
\]

\#-mark \( \beta \) in the mdr slot of proplet \( \alpha \)

To realize the punctuation sign and to enable a potential continuation to a next proposition (NLC Chap. 12), the navigation travels back to the predicate with \( A \uparrow V \):

3.6.19 **Navigating with \( A \uparrow V \)** from *loud* back to *snore* (arc 8)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \quad \text{content level} \\
\begin{array}{l}
\text{adj: } \beta \\
\text{mdr: } \alpha \\
\text{prn: } K
\end{array}
& \quad \begin{array}{l}
\text{sur: } \text{lexverb}(\alpha) \\
\text{verb: } \alpha \\
\text{mdr: } X \beta Y \\
\text{prn: } K
\end{array}
\end{align*}
\]

Using the \textit{cat} value \( \text{decl} \) of *snore*, lexverb realizes the period.

We follow standard procedure and conclude the linguistic analysis of the example with point 7 on the to-do list [1.5.1]. Extending DBS.11 to an example class with an intrapropositional noun coordination required the definition of the four hear mode operations \( N \times N \) (h28), \( N_1 \times N \) (h29), \( N \sim \text{CNJ} \) (h33), and \( \text{CNJ} \cup N \) (h39), and the two speak mode operations \( N \rightarrow N \) (s21) and \( N \leftarrow N \) (s22). Counting the suspension in line 2 and its compensation in lines 3a and 3b, the hear mode derivation used seven operation applications, and the speak mode derivation used eight:
3.6.20 COMPARING HEAR AND SPEAK MODE SURFACE SEGMENTATION

**hear mode:**

1 2 3a 3b 4 5 6

Fido $\times$ Tucker $\sim$ and $\cup$ $\times$ Buster $\times$ snored $\times$ loudly $\cup$ .

**speak mode:**

1 2 3 4 5

V/N Fido N→N Tucker N→N and_Buster N←V N←N

6 7 8

N/V snored V↓A loudly A↑V .

In the hear mode, adding a conjunct requires one operation application (excepting the conjunction), but its speak mode production requires two.
4. Valency Positions

This chapter analyzes valency structures which go beyond simple one-place, two-place, and three-place main verbs. They are main verb constructions taking a prepositional object (Sect. 4.1), a discontinuous elementary adnominal (Sect. 4.2), a discontinuous bare preposition (Sect. 4.3), and an infinitival object (Sect. 4.4) as argument, as well as copula constructions taking a second noun (Sect. 4.5) or an adnominal (Sect. 4.6).

4.1 DBS.13: Verb Taking a Prepositional Object

A verb taking a prepositional phrase as an obligatory argument is put, as in Julia put the flowers on the table: without the prepositional object on the table it would be grammatically incomplete. With other verbs, for example eat, the same prepositional phrase may be used as an optional modifier. Thus, Fido ate the bone on the table (3.2.2) would not become grammatically incomplete if the prepositional phrase were omitted.

The example of put as a three place verb taking two obligatory nouns and one obligatory prepositional object as arguments has the following content:

4.1.1 CONTENT OF THE SURFACE Julia put the flowers on the table.

---

1 Optional modifiers are sometimes called *adjuncts*. This term expresses their optional nature, but not the semantic relation of structure. A more mundane reason for preferring the modifier vs. argument distinction is the difference in their initial letters, m vs. a, which are used as superscripts in the operation names 3.2.7 vs. 4.1.6. The initial letters in the adjunct vs. argument and in the obligatory vs. optional dichotomies, in contrast, are the same and therefore unsuitable for marking the distinction.
The connection between the verb and the prepositional phrase serving as an argument is coded by a combination of two methods. The first is common to optional modifiers (3.2.2) and obligatory arguments (4.1.1), and consists in cross-copying the core values into the mdd and the mdr slots. The second is specific to prepositional phrases serving as an obligatory argument; it is based on the presence of an mdr valency position in the third cat slot of the verb and canceling this valency position by #-marking. For example, the lexical feature [cat: n’ a’ mdr’ v] of put ends up in [4.1.1] as [cat: #n’ #a’ #mdr’ decl].

Consider the hear mode derivation of [4.1.1] in canonical DBS graph format:
The crucial step is in line 4. It consists in (i) the cross-copying between the verb and the preposition, and (ii) the concomitant canceling of the valency position in the cat slot of put (not indicated graphically, but see line 5).

The three preceding and the three following derivation steps resemble 3.2.3 in that they use the same operations in the same order:

4.1.3 CROSS-COPYING Julia and put with SBJ × PRD (line 1)
4.1.4 CROSS-COPYING put and the with PRD×OBJ (LINE 2)

**PRD×OBJ**

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: β</td>
<td>sur: the</td>
</tr>
<tr>
<td>cat: #X N' Y γ</td>
<td>noun: n_1</td>
</tr>
<tr>
<td>arg: Z</td>
<td>cat: #n' a' mdr' v</td>
</tr>
<tr>
<td>prn: K</td>
<td>sem: past</td>
</tr>
</tbody>
</table>

\[ \Rightarrow \]

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: β</td>
<td>sur: flowers</td>
</tr>
<tr>
<td>cat: #X N' Y γ</td>
<td>noun: flower</td>
</tr>
<tr>
<td>arg: Z α</td>
<td>cat: #n' a' mdr' v</td>
</tr>
<tr>
<td>prn: K</td>
<td>sem: def</td>
</tr>
</tbody>
</table>

Up to this point, the hear mode derivation has interpreted Julia put the flowers.

The next operation to apply is new PRD×PREP\(^a\). The superscript \(^a\) stands for (obligatory) argument, in contradistinction to the superscript \(^m\) for (optional) modifier. Like PRD×PREP\(^m\) (3.2.7), PRD×PREP\(^a\) connects the finite verb form and the preposition (as the beginning of a prepositional phrase) by cross-copying the core values into the respective mdd and mdr slots. Unlike PRD×PREP\(^m\), PRD×PREP\(^a\) requires an mdr' valency position in the cat slot of the verb and cancels the valency position by #-marking.

These details are coded in the transition from the input patterns to the output patterns of the PRD×PREP\(^a\) operation:
4.1.6 CROSS-COPYING put AND on WITH PRD×PREP (line 4)

The feature \([\text{cat}: \#X \text{ mdr'} v]\) of the first input pattern requires a verb taking a modifier as its last argument. The second input pattern requires a preposition, as shown by the core attribute noun and the cat value advn. In the two output proplets, the core value of the preposition is cross-copied into the mdr slot of the verb and the core value of the verb into the mdd slot of the preposition. The first output pattern cancels the valency position mdr' with the feature \([\text{cat}: \#X \#\text{mdr'} v]\).

The next word is the definite article, which continues the preposition into a prepositional phrase. The article is absorbed into the preposition by simultaneous substitution replacing \(n_2\) with \(n_3\) and by fusing the cat values of the two input proplets in the output proplet, similar to 3.2.8.

4.1.7 ABSORBING the INTO \(n_2\) WITH PREP\(\cup\)NP (line 5)

This operation may also combine a preposition and an elementary noun, as in Paris or to you. Here, however, the addition of a determiner requires the addition of a noun, with or without adnominal modifiers.
The input continues with the next word **table** and **DET∪CN** applies next:

### 4.1.8 ABSORBING **table** INTO \( n_3 \) WITH **DET∪CN** (line 6)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
<th>Agreement conditions as in 2.2.4</th>
</tr>
</thead>
</table>
| **DET∪CN (h37)** | **sur:** noun: \( n_3 \)  
   cat: X CN’ NP  
   sem: Y  
   prn: K | **sur:** noun: **table**  
   cat: X NP  
   sem: Y Z  
   prn: K |
| **sur:** noun: \( n_3 \)  
   cat: adnv nn’ np  
   sem: on def sg  
   mdd: put  
   mdr:  
   nc:  
   pc:  
   prn: 19 | **sur:** noun: **table**  
   cat: adnv snp  
   sem: on def sg  
   mdd: put  
   mdr:  
   nc:  
   pc:  
   prn: 19 |

Simultaneous substitution replaces all instances of \( n_3 \) with **table**.

Having completed the prepositional phrase, it remains to add the punctuation, here period, with standard **S∪IP**:

### 4.1.9 ABSORBING . INTO **put** WITH **S∪IP** (line 7)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
<th>Agreement conditions as in 2.1.3</th>
</tr>
</thead>
</table>
| **S∪IP (h13)** | **sur:** verb: \( \beta \)  
   cat: #X’ VT  
   prn: K | **sur:** verb: V\(_n\)  
   cat: VT’ SM  
   prn: K |
| **sur:** verb: put  
   cat: #n‘ #a’ #mdr’ v  
   sem: past  
   arg: (person x) flower  
   mdr: table  
   nc:  
   pc:  
   prn: 19 | **sur:** verb: v\(_1\)  
   cat: v decl  
   prn: |

Comparison of Fido ate the bone on the table (3.2.2) and of Julia put the flowers on the table (4.1.1) shows that the semantic distinction between a prepositional phrase as an optional modifier and an obligatory argument originates in the lexical definition of the respective verbs, here *eat* vs. *put*: they differ in the list of valency positions in their cat feature. In all other respects, the adverbial interpretations in Sects. 3.2 and 4.1 are analogous.

Having completed the hear mode derivation, we follow standard procedure (1.5.1 4) by beginning the speak mode analysis with a DBS graph analysis:
4.1.10 **Graph analysis underlying production of**

(i) **SRG (semantic relations graph)**

(ii) **signature**

(iii) **NAG (numbered arcs graph)**

(iv) **surface realization**

The arc numbering is breadth first (Sect. 1.4). For simplicity, the distinction between a prepositional phrase used (i) as an optional modifier (3.2.11) and (ii) as an obligatory argument (4.1.6) is not reflected graphically.

The word form production of the example may be summarized as follows:

4.1.11 **Sequence of operation names and surface realizations**

arc 1: \( V/\text{N} \) from \( \text{put} \) to \( \text{julia} \) \( \text{Julia} \)

arc 2: \( N/\text{V} \) from \( \text{julia} \) to \( \text{put} \) \( \text{put} \)

arc 3: \( V/\text{N} \) from \( \text{put} \) to \( \text{flower} \) \( \text{the\_flowers} \)

arc 4: \( N/\text{V} \) from \( \text{flower} \) to \( \text{put} \) \( \text{the\_flowers} \)

arc 5: \( V\downarrow\text{N} \) from \( \text{put} \) to \( \text{table} \) \( \text{on\_the\_table} \)

arc 6: \( N\uparrow\text{V} \) from \( \text{table} \) to \( \text{put} \).

We continue with point 5 on the to-do list (1.5.1), beginning with arc 1:

4.1.12 **Navigating with \( V/\text{N} \) from \( \text{put} \) to \( \text{julia} \) (arc 1)**

Any distinction, e.g. between singular and plural, may be reflected in a DBS graph like 4.1.10 if so desired, for example, by using different colors for nodes or by using different line styles.
4.1.13 Navigating with N/V from *julia* back to *put* (arc 2)

\[
\begin{pmatrix}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K
\end{pmatrix} \Rightarrow \begin{pmatrix}
\text{sur: lexverb(\(\hat{\alpha}\))} \\
\text{verb: } \alpha \\
\text{arg: } #\beta X \\
\text{prn: } K
\end{pmatrix}
\]

#-mark \(\alpha\) in the fnc slot of proplet \(\beta\)

\[
\begin{pmatrix}
\text{sur: julia} \\
\text{noun: (person x)} \\
\text{cat: snp} \\
\text{sem: nm f} \\
\text{fnc: put} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 19}
\end{pmatrix} \uparrow \downarrow \begin{pmatrix}
\text{sur: put} \\
\text{verb: put} \\
\text{cat: } #n' #a' #mdr' decl \\
\text{sem: past} \\
\text{arg: } #(person x) \text{ flower} \\
\text{mdr: table} \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 19}
\end{pmatrix}
\]

4.1.14 Navigating with V\(\backslash\)N from *put* to *flower* (arc 3)

\[
\begin{pmatrix}
\text{verb: } \alpha \\
\text{arg: } #X \beta Y \\
\text{prn: } K
\end{pmatrix} \Rightarrow \begin{pmatrix}
\text{sur: lexnoun(\(\hat{\beta}\))} \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K
\end{pmatrix}
\]

#-mark \(\beta\) in the arg slot of proplet \(\alpha\)

\[
\begin{pmatrix}
\text{sur: the_flowers} \\
\text{noun: flower} \\
\text{cat: pnp} \\
\text{sem: def pl} \\
\text{fnc: #put} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 19}
\end{pmatrix} \uparrow \downarrow \begin{pmatrix}
\text{sur: put} \\
\text{verb: put} \\
\text{cat: } #n' #a' #mdr' decl \\
\text{sem: past} \\
\text{arg: } #(person x) \text{ flower} \\
\text{mdr: table} \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 19}
\end{pmatrix}
\]

4.1.15 Navigating with N\(\backslash\)V from *flower* back to *put* (arc 4)

\[
\begin{pmatrix}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K
\end{pmatrix} \Rightarrow \begin{pmatrix}
\text{sur: lexverb(\(\hat{\alpha}\))} \\
\text{verb: } \alpha \\
\text{arg: } #X #\beta Y \\
\text{prn: } K
\end{pmatrix}
\]

Z is #-marked or NIL

\[
\begin{pmatrix}
\text{sur: } \\
\text{noun: flower} \\
\text{cat: pnp} \\
\text{sem: def pl} \\
\text{fnc: #put} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn 19}
\end{pmatrix} \uparrow \downarrow \begin{pmatrix}
\text{sur: put} \\
\text{verb: put} \\
\text{cat: } #n' #a' #mdr' decl \\
\text{sem: past} \\
\text{arg: } #(person x) \text{ flower} \\
\text{mdr: table} \\
\text{nc: } \\
\text{pc: } \\
\text{prn 19}
\end{pmatrix}
\]
The next step navigates from the verb to the prepositional phrase with \( V \downarrow N \), similar to 3.2.17. The operation ignores the difference between an obligatory argument vs. an optional modifier by omitting the \texttt{cat} feature in its input.

### 4.1.16 Navigating with \( V \downarrow N \) from put to table (arc 5)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \\
\hline
\begin{array}{l}
\text{verb: } \alpha \\
\text{mdr: } X \beta Y \\
\text{prn: } K
\end{array} & \begin{array}{l}
\text{sur: lexnoun(}\hat{\beta}) \\
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{array} & \text{#-mark } \beta \text{ in the arg slot of proplet } \alpha
\end{array}
\]

\[
\begin{array}{l}
\text{sur: } \text{on_the_table} \\
\text{noun: table} \\
\text{cat: adnp} \\
\text{sem: on def sg} \\
\text{mdd: put} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 19}
\end{array}
\]

Lexnoun uses the features [noun: table] and [sem: on def sg].

Finally, the navigation returns to the top predicate to realize the punctuation mark period, analogous to 3.2.18.

### 4.1.17 Navigating with \( N \uparrow V \) from table back to put (arc 6)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \\
\hline
\begin{array}{l}
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{array} & \begin{array}{l}
\text{sur: lexverb(}\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{mdr: } X \beta Y \\
\text{prn: } K
\end{array} & \\
\hline
\begin{array}{l}
\text{sur: } \\
\text{noun: table} \\
\text{cat: adnp} \\
\text{sem: on def sg} \\
\text{mdd: put} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: 19}
\end{array}
\end{array}
\]

Verbs like put may also take elementary modifiers as their obligatory argument, as in Julia put the flowers down. Clausal adverbial modifiers (Sect. 3.5), in contrast, seem to be excluded.

Extending DBS.12 to a class of examples with an obligatory modifier argument required the definition of the hear mode operations \( \text{PRD} \times \text{PREP}^n \) (126) and \( \text{PREP} \cup \text{NP} \) (138). The hear mode derivation used seven operation applications, and the speak mode derivation used six:
4.1.18 Comparing Hear and Speak Mode Surface Segmentation

Hear mode:
1 2 3 4 5 6 7
Julia × put × the ∪ flowers × on ∪ the ∪ table ∪ .

Speak mode:
1 2 3 4 5 6
V\N Julia N/V put V\N the_flowers N\V V\N on_the_table N\V .

The hear mode derivation runs straight through. The breadth first numbering of the speak mode operations corresponds to the arcs traversed in 4.1.10; there is one empty traversal.

4.2 DBS.14: Verb Taking a Discontinuous Adnominal

The artificial intelligence aspect of DBS aims at reconstructing the semantics of content words (concepts) in the form of the agent’s elementary recognition and action procedures, defined declaratively as types and implemented procedurally as tokens (grounding, FoCL Sect. 3.3, Barsalou 2008). The AI aspect is complemented by the linguistic aspect, which aims at classifying content words in terms of selectional constellations, based on an RMD corpus (CLaTR Sect. 15.3).3

The AI and the linguistic approach complement each other. The computational reconstruction of a concept as a recognition or action procedure may be applied to its lexical class as a selectional constellation. Conversely, the grammatical properties of a selectional constellation class may help in the computational implementation of the concepts in that class as software constructs.

Classes of selectional constellations, based on the bare co-occurrence facts distilled from a corpus, may also be used for traditional linguistic reasoning. For example, make in {make...happy} seems to be restricted to such “mood” adnominals as {angry, happy, pensive, proud, tired, sad, wary, ...}. Furthermore, the constructions illustrated by put...on_a_table (Sect. 4.1) vs. make...happy shown below have in common that (i) they are discontinuous and that (ii) the obligatory modifier is an argument, and not a part of the verb.

Let us follow standard procedure and begin with point 1 on the to-list 1.5.1:

3 A selectional constellation may specify, for example, that the mdr’ valency position of put in 4.1.10 may be canceled alternatively by a prepositional phrase, as in put ... on the table, or a lone preposition, as in put ... down, whereby the choice of the preposition depends on the semantics of the verb, reflected in different collocational conventions and listed for the class as a variable restriction.
4.2.1 CONTENT OF THE SURFACE  The letter made Mary happy.

The content resembles 4.1.1 except that the modifier here is an elementary adnominal instead of a prepositional phrase.

The similarity appears also in the time-linear hear mode derivation:

4.2.2 GRAPHICAL HEAR MODE DERIVATION OF THE CONTENT 4.2.1

The letter made Mary happy.
4. Valency Positions

Valency Positions

The derivation is two lines shorter than 4.1.2 because the modifier argument is elementary happy rather than phrasal on the table.

Following standard procedure, we continue with point 3 on the to-do list 1.5.1, i.e. the complete sequence of explicit hear mode applications, starting with standard DETCN, SBJ×PRD, and PRD×OBJ:

4.2.3 Absorbing letter into n₁ with DETCN (line 1)

The next word provided by automatic word form recognition is make. It matches the second input pattern of SBJ×PRD, which finds letter at the now front matching its first input pattern and applies as follows:

4.2.4 Cross-copying letter and made with SBJ×PRD (line 2)
4.2.5 **Cross-copying `make` and Mary with PRD×OBJ (Line 3)**

\[
\begin{array}{c|c|c}
\text{content level} & \text{pattern level} & \text{content level} \\
\hline
\text{sur: } & \text{sur: } & \text{sur: } \\
\text{noun: letter} & \text{noun: letter} & \text{noun: letter} \\
\text{cat: snp} & \text{verb: make} & \text{cat: snp} \\
\text{sem: def sg} & \text{cat: n' a' mdr' v} & \text{sem: def sg} \\
\text{fnc:} & \text{sem: past} & \text{fnc:} \\
\text{mdr:} & \text{arg:} & \text{mdr:} \\
\text{nc:} & \text{nc:} & \text{nc:} \\
\text{pc:} & \text{pc:} & \text{pc:} \\
\text{prn: 20} & \text{prn: 20} & \text{prn: 20} \\
\end{array}
\]

Agreement conditions as in 2.3.4.

The obligatory modifier argument *happy* is added by the new operation PRD×ADN, as an alternative to PRD×PREP\(^a\) (4.1.6):

4.2.6 **Cross-copying `make` and happy with PRD×ADN (Line 4)**

\[
\begin{array}{c|c|c|c}
\text{pattern level} & \text{content level} & \text{content level} & \text{pattern level} \\
\hline
\text{verb: } & \text{sur: } & \text{sur: } & \text{verb: } \\
\text{\#X' N' Y \gamma} & \text{adj: } & \text{adj: } & \text{adj: } \\
\text{cat: CN' N} & \text{\#X' \#N' \gamma \alpha} & \text{\#X' \#\gamma' \alpha} & \text{\#X' \#\gamma' \alpha} \\
\text{arg: Z} & \text{cat: ADN} & \text{cat: ADN} & \text{cat: ADN} \\
\text{pc:} & \text{mdr:} & \text{mdr:} & \text{mdr:} \\
\text{nc:} & \text{nc:} & \text{nc:} & \text{nc:} \\
\text{mdr:} & \text{mdr:} & \text{mdr:} & \text{mdr:} \\
\text{nc:} & \text{nc:} & \text{nc:} & \text{nc:} \\
\text{pc:} & \text{pc:} & \text{pc:} & \text{pc:} \\
\text{prn: 20} & \text{prn: 20} & \text{prn: 20} & \text{prn: 20} \\
\end{array}
\]

\(\gamma \in \{\text{mdr'}, \text{be}\}. \) If \(\gamma = \text{mdr'}\), then ADN = adn; otherwise ADN \(\in \{\text{adn, adv}\}\)

The added modifier is an adnominal rather than an adverbial because it must be, for example, happy and not happily. Analogous to PRD×PREP\(^a\) (1.26).
PRD × ADN (i) cross-copies the core values of the verb and the adnominal into the respective mdr and mdd slots, and (ii) cancels the mdr′ valency position in the third cat slot of the verb, thus characterizing happy as obligatory.

The hear mode derivation concludes with S∪IP absorbing the period into the finite verb:

4.2.7 ABSORBING • INTO make WITH S∪IP (line 5)

Following standard procedure, let us continue with point 4 on the to-do list 1.5.1, i.e. the DBS graph showing the semantic relations of structure, the navigation, and the surface realization of the speak mode:

4.2.8 GRAPH ANALYSIS UNDERLYING PRODUCTION OF 4.2.1

Compared to the graph 4.1.10 for Julia put the flowers on the table, the signature node representing the modifier argument is A rather than N. The respective NAGs show the same breadth first arc numbering.
We continue with point 5 on the to-do list \[5.1\] i.e. the list of speak mode operation names with surface realizations:

### 4.2.9 Sequence of Operation Names and Surface Realizations

<table>
<thead>
<tr>
<th>Arc</th>
<th>Name</th>
<th>Operation</th>
<th>Surface Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V/N</td>
<td>from make to letter</td>
<td>The_letter</td>
</tr>
<tr>
<td>2</td>
<td>N/V</td>
<td>from letter to make</td>
<td>made</td>
</tr>
<tr>
<td>3</td>
<td>V/N</td>
<td>from make to mary</td>
<td>Mary</td>
</tr>
<tr>
<td>4</td>
<td>N/V</td>
<td>from mary to make</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>V\downarrow A</td>
<td>from make to happy</td>
<td>happy</td>
</tr>
<tr>
<td>6</td>
<td>A\uparrow V</td>
<td>from happy to make</td>
<td></td>
</tr>
</tbody>
</table>

Given that a name and a determiner\-noun combination are represented alike as a single proplet, the first four derivation steps resemble those in \[3.2.13-3.2.16 and 4.1.12-4.1.15\]

#### 4.2.10 Navigating with V/N from make to letter (arc 1)

\[
\begin{align*}
\text{pattern level} & : \\
\verb: \alpha & , \text{arg: } \beta \ \text{X} \\
\text{prn: } & K \\
\Rightarrow & : \\
\text{content level} & : \\
\text{sur: } & \text{lexnoun(\(\hat{\beta}\))} \\
\text{verb: } & \alpha \\
\text{arg: } & \#X \\
\text{noun: } & \beta \\
\text{prn: } & K \\
\hat{\beta} & \text{in the arg slot of proplet } \alpha
\end{align*}
\]

#### 4.2.11 Navigating with N/V from letter back to make (arc 2)

\[
\begin{align*}
\text{pattern level} & : \\
\text{noun: } & \beta \\
\text{fnc: } & \alpha \\
\text{mdr: } & Z \\
\text{prn: } & K \\
\Rightarrow & : \\
\text{content level} & : \\
\text{sur: } & \text{lexverb(\(\hat{\alpha}\))} \\
\text{verb: } & \alpha \\
\text{arg: } & \#X \\
\text{nuc: } & \hat{\beta} \\
\text{prn: } & K \\
\hat{\alpha} & \text{in the fnc slot of proplet } \beta
\end{align*}
\]

\(Z\) is NIL, or elementary and \#-marked

\[
\begin{align*}
\text{sur: } & \text{letter} \\
\text{cat: } & \text{snp} \\
\text{sem: } & \text{def sg} \\
\text{fnc: } & \text{make} \\
\text{mdr: } & \text{happy} \\
\text{nc: } & \text{pc:} \\
\text{prn: } & 20
\end{align*}
\]
4.2.12 NAVIGATING WITH $\text{V}N$ FROM make TO (person $x$) (arc 3)

\[ \text{V}N \begin{pmatrix} \verb: \alpha \\
\text{arg: } \#X \beta Y \\
\text{prn: } K \end{pmatrix} \Rightarrow \begin{pmatrix} \text{sur: lexnoun(\hat{\beta})} \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K \end{pmatrix} \]

#-mark $\beta$ in the arg slot of proplet $\alpha$

Up to this point, the speak mode navigation has realized The letter made Mary. The return to the predicate with standard $N\text{V}$ provides access to the continuation feature [mdr: happy] in the goal proplet:

4.2.13 NAVIGATING WITH $\text{N}V$ FROM mary BACK TO make (arc 4)

\[ \text{N}V \begin{pmatrix} \text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K \end{pmatrix} \Rightarrow \begin{pmatrix} \text{sur: lexverb(\hat{\alpha})} \\
\verb: \alpha \\
\text{arg: } \#X \beta Y \\
\text{prn: } K \end{pmatrix} \]

$Z$ is #-marked or NIL

The pattern of $N\text{V}$ ignores the continuation feature [mdr: happy] in the output proplet by omitting the mdr attribute.

For the next operation $V_A$, however, the feature [mdr: $\#X \beta Y$] is an essential part of the start pattern. In contradistinction to $V\text{N} (4.1.16)$, the core attribute of the goal pattern is adj instead of noun. Using the mdr value happy in the start proplet make, the following operation application reaches the modifier argument:
4.2.14 Navigating with $V \downarrow A$ from *make* to *happy* (arc 5)

For realizing *happy*, lexadj uses the core value, the cat value `adn`, and the sem value `pad` (for adjective in the positive).

A return to the verb with existing $A \uparrow V$ (e.g. 3.1.14), as the inverse of $V \downarrow A$, completes the speak mode derivation:

4.2.15 Navigating with $A \uparrow V$ from *make* back to *happy* (arc 6)

Without any untraversed continuation values in the *make* proplet, lexverb uses the cat value `decl` to realize the period.

Following standard procedure, we conclude the section with point 7 on the to-do list. Extending DBS.13 to an example class with a verb taking an obligatory discontinuous adnominal required the definition of the hear mode operation $PRD \times ADN$ (h42). The hear mode derivation used five operation applications, and the speak mode derivation used six:
4.2.16 Comparing Hear and Speak Mode Surface Segmentation

**Hear mode:**

1 2 3 4 5

The ∪ letter × made × Mary × happy ∪ .

**Speak mode:**

1 2 3 4 5 6

V/N The_letter N/V made V\N Mary N\V V↓A happy A↑V .

The hear mode derivation runs straight through. The numbering of the speak mode operations corresponds to the breadth first arc numbering in 4.2.8.

4.3 DBS.15: Verb Taking a Discontinuous Bare Preposition

While make...happy consists of two content words, there are also discontinuous structures consisting of a content and a function word, for example dig...up.

Called “discontinuous element” in English grammar, up raises the question of whether it should be treated as an argument of the verb, like happy, or as a part of the verb, called “separable prefix.” In German grammar, verbs with a separable prefix (trennbare Verben) are written as one in lexical entries, e.g. ausgraben (up_dig), and in subordinate clauses, e.g. als Fido den Knochen ausgrub (when Fido up_dug the bone), but as two separate surfaces in main clauses, e.g. Fido grub den Knochen aus (Fido dug the bone up).

In line with our analysis of prepositional (Sect. 4.1) and adnominal (Sect. 4.2) arguments, the following analysis of English treats the discontinuous element up as an argument, and not as a part of, the verb dig. As an argument, it cancels a corresponding valency position in the cat feature of the verb (NEWCAT 1986). As a function word, the preposition is absorbed into the verb.

The valency position for a preposition is represented by bp′, for “bare preposition.” Its fillers are defined as the restriction on a variable, called BP, which is assigned to the lexical entry of the verb in question and allows several possible prepositions as fillers. For example, the BP-set of put includes up, down, in, out, on, ... . Using the specialized valency slot bp′ instead of general mdr′ strengthens the relation between a verb and its discontinuous elements without abandoning their role as an argument.

Consider the following content with the verb dig taking the bare preposition up as a filler:

---

\footnote{At least since 1960 (Bar-Hillel 1964), this construction has played a crucial role in the question of how to accommodate “constituent structure” in context-free PS Grammar (FoCL p. 160; CLaTR Sect. 12.2).}
4.3 CONTENT OF THE SURFACE *Fido dug the bone up.*

The absorbed preposition *up* serving as the discontinuous element of the example is preserved as the initial value in the *sem* slot of *dig*.

The time-linear, surface compositional hear mode derivation of this content is as follows (point 2 on the to-do list 1.5.1):

4.3.2 GRAPHICAL HEAR MODE DERIVATION OF THE CONTENT 4.3.1

<table>
<thead>
<tr>
<th>unanalyzed surface</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fido dug the bone up</strong>.</td>
</tr>
<tr>
<td>automatic word form recognition</td>
</tr>
<tr>
<td><em>sur: Fido</em></td>
</tr>
<tr>
<td>noun: (dog x)</td>
</tr>
<tr>
<td>sem: nm m</td>
</tr>
<tr>
<td>fnc: dig</td>
</tr>
<tr>
<td>prn: 21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>syntactic–semantic parsing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>noun: (dog x)</td>
</tr>
<tr>
<td>cat: snp</td>
</tr>
<tr>
<td>sem: nm m</td>
</tr>
<tr>
<td>fnc: dig</td>
</tr>
<tr>
<td>prn: 21</td>
</tr>
</tbody>
</table>

| 2 | *sur: Fido* | *sur: dug* | *sur: the* | *sur: bone* | *sur: up* |
| noun: (dog x) | verb: dig | noun: n_1 | noun: bone | noun: n_2 | |
| cat: snp | cat: n’ a’ bp’ v | cat: nm’ np | cat: sn | cat: advn |
| sem: nm m | sem: past | sem: def | sem: sg |
| fnc: dig | arg: (dog x) bone | fnc: | mdd: |
| prn: 21 | prn: 21 | prn: | prn: |

| 3 | *sur: Fido* | *sur: dug* | *sur: bone* | *sur: up* |
| noun: (dog x) | verb: dig | noun: n_1 | noun: bone | noun: n_2 |
| cat: snp | cat: n’ a’ bp’ v | cat: nm’ np | cat: sn |
| sem: nm m | sem: past | sem: def | sem: sg |
| fnc: dig | arg: (dog x) bone | fnc: | mdd: |
| prn: 21 | prn: 21 | prn: | prn: |

| 4 | *sur: Fido* | *sur: dug* | *sur: bone* | *sur: up* |
| noun: (dog x) | verb: dig | noun: bone | noun: bone |
| cat: snp | cat: n’ a’ bp’ v | cat: nm’ np | cat: advn |
| sem: nm m | sem: past | sem: def | sem: up |
| fnc: dig | arg: (dog x) bone | fnc: | mdd: |
| prn: 21 | prn: 21 | prn: | prn: |
The bare preposition up is absorbed in line 4. It fills the valency position bp′ in the third cat slot of dig and leaves its sem value up in the initial sem slot of the verb. As a function word, up is absorbed into dig without leaving behind a separate proplet representing a modifier argument in the resulting content, unlike table in 4.1.1 and happy in 4.2.1.

Let us continue with point 3 on the to-do list 4.1.1, i.e. the sequence of explicit hear mode operation applications which derive 4.3.1. The first operation to apply is SBJ×PRD:

### 4.3.3 Cross-copying Fido and dug with SBJ×PRD (line 1)

For SBJ×PRD to apply, (i) the cat feature of the first input proplet must match the cat feature in the first input pattern (vertical) and (ii) the cat value corresponding to NP in the first input proplet must agree with the cat value corresponding to NP′ in the second input pattern (horizontal). The same principle of canceling valency positions in their order from left to right in the verb’s cat slot is used in the following application of PRD×OBJ:
4.3.4 CROSS-COPYING *dig* AND THE WITH PRD × OBJ (LINE 2)

\[
\begin{array}{ccc}
\text{PRD} \times \text{OBJ} (\text{line 2}) & & \text{PRD} \times \text{OBJ} (\text{line 2}) \\
\begin{array}{|c|c|c|}
\hline
\text{verb: } \beta & \text{noun: } \alpha & \text{verb: } \beta \\
\text{cat: } #X' \text{ N' Y } \gamma & \text{cat: } \text{CN' N} & \text{cat: } \text{CN' N} \\
\text{arg: } Z & \text{fnc: } \alpha & \text{fnc: } \beta \\
\text{prn: } K & \text{prn: } Z & \text{prn: } K \\
\hline
\end{array} \\
\Rightarrow \\
\begin{array}{|c|c|c|}
\hline
\text{sur: } \text{dig} & \text{noun: } n_1 & \text{sur: } \text{dig} \\
\text{cat: } #n' \text{ a' bp' v} & \text{cat: } \text{np} & \text{cat: } \text{np} \\
\text{sem: past} & \text{sem: def} & \text{sem: def} \\
\text{arg: } (\text{dog } x) & \text{fnc: } & \text{fnc: } \\
\text{mdr: } & \text{mdr: } & \text{mdr: } \\
\text{nc: } & \text{nc: } & \text{nc: } \\
\text{pc: } & \text{pc: } & \text{pc: } \\
\text{prn: } 21 & \text{prn: } 21 & \text{prn: } 21 \\
\hline
\end{array}
\end{array}
\]

The object is completed with DET ∪ CN absorbing bone into the determiner:

4.3.5 ABSORBING bone INTO n_1 WITH DET ∪ CN (line 3)

\[
\begin{array}{ccc}
\text{DET} \cup \text{CN} (\text{line 3}) & & \text{DET} \cup \text{CN} (\text{line 3}) \\
\begin{array}{|c|c|c|}
\hline
\text{noun: } N \text{ n} & \text{cat: } \text{CN} & \text{noun: } \alpha \\
\text{cat: } \text{X CN' NP} & \text{sem: } \text{Z} & \text{cat: } \text{X NP} \\
\text{prn: } K & \text{prn: } & \text{prn: } K \\
\hline
\end{array} \\
\Rightarrow \\
\begin{array}{|c|c|c|}
\hline
\text{sur: } \text{bone} & \text{noun: } \text{bone} & \text{sur: } \text{bone} \\
\text{cat: } \text{sn} & \text{cat: } \text{sp} & \text{cat: } \text{sp} \\
\text{sem: sg} & \text{sem: def sg} & \text{sem: def sg} \\
\text{fnc: } & \text{fnc: } & \text{fnc: } \\
\text{mdr: } & \text{mdr: } & \text{mdr: } \\
\text{nc: } & \text{nc: } & \text{nc: } \\
\text{pc: } & \text{pc: } & \text{pc: } \\
\text{prn: } 21 & \text{prn: } 21 & \text{prn: } 21 \\
\hline
\end{array}
\end{array}
\]

At this point, only bp' (for bare preposition) remains as an uncanceled valency position in the cat slot of *dig*.

As the next word, *up* is provided by automatic word form recognition and absorbed into *dig* by PRD ∪ BP:

4.3.6 ABSORBING up INTO dig WITH PRD ∪ BP (line 4)

\[
\begin{array}{ccc}
\text{PRD} \cup \text{BP} (\text{line 4}) & & \text{PRD} \cup \text{BP} (\text{line 4}) \\
\begin{array}{|c|c|c|}
\hline
\text{verb: } \alpha & \text{noun: } N \text{ n} & \text{verb: } \alpha \\
\text{cat: } #X' \beta' v & \text{cat: } \text{adv} & \text{cat: } #X' \beta' v \\
\text{sem: } \beta & \text{sem: } \beta & \text{sem: } \beta \\
\text{prn: } K & \text{prn: } K & \text{prn: } K \\
\hline
\end{array} \\
\Rightarrow \\
\begin{array}{|c|c|c|}
\hline
\text{sur: } \text{bone} & \text{noun: } \text{bone} & \text{sur: } \text{bone} \\
\text{cat: } \text{sp} & \text{cat: } \text{sp} & \text{cat: } \text{sp} \\
\text{sem: def sg} & \text{sem: def sg} & \text{sem: def sg} \\
\text{fnc: } & \text{fnc: } & \text{fnc: } \\
\text{mdr: } & \text{mdr: } & \text{mdr: } \\
\text{nc: } & \text{nc: } & \text{nc: } \\
\text{pc: } & \text{pc: } & \text{pc: } \\
\text{prn: } 21 & \text{prn: } 21 & \text{prn: } 21 \\
\hline
\end{array}
\end{array}
\]
4. Valency Positions

where \( \beta \in \{\text{in, out, up, ...}\} \)

Though \( \text{PRD} \cup \text{BP} \) resembles \( \text{PRD} \times \text{ADN} \) (4.2.6) and \( \text{PRD} \times \text{PREP}^a \) (4.1.6) in that it adds a modifier argument, it differs from them in that it absorbs the discontinuous element as a function word, instead of cross-copying between content words.

With all three valency positions in the \text{cat} slot of \text{dig} canceled, it remains to add the punctuation sign period with standard \( \text{S} \cup \text{IP} \):

4.3.7 Absorbing \( \bullet \) INTO \text{dig} WITH \( \text{S} \cup \text{IP} \) (line 5)

This completes the hear mode derivation of a discontinuous element construction. The bare preposition \( \text{up} \) is shown as the first value in the \text{sem} slot of \text{dig}. In accordance with surface compositionality, there is a lexical \( \text{up} \) proplet in the input of 4.3.6 but as a function word it is absorbed. This is in contradistinction to the adnominal \text{happy} as the modifier argument of \text{make} in Sect. 4.2 and the prepositional phrase \text{on the table} as the modifier argument of \text{put} in Sect. 4.1, which remain as proplets in the resulting contents.

Let us follow standard procedure and continue with point 4 on the to-do list 1.5.1, i.e. the DBS graph showing the semantic relations of structure and the surface realization in the speak mode:
4.3.8 **Graph Analysis Underlying Production of**

(i) **SRG (semantic relations graph)**

```
   dig
   /   \
  fido  bone
```

(ii) **Signature**

```
   V
   / \
  N   N
```

(iii) **NAG (numbered arcs graph)**

```
1  dig  2
   \
  fido
3  bone
```

(iv) **Surface Realization**

```
Fido dug the_bone up_.
```

The graph differs from 4.1.10 for *Julia put the flowers on the table* and from 4.2.8 for *The letter made Mary happy* in that there is neither a $V \downarrow N$ nor a $N \uparrow V$ relation. Instead, lexverb realizes $up_-$ from the $\text{sem}$ and $\text{cat}$ values of the verb in arc 4.

We continue with point 5 on the to-do list 1.5.1, i.e. the list of speak mode operation names with surface realizations:

### 4.3.9 Sequence of Operation Names and Surface Realizations

- **arc 1:** $V/N$ from *dig* to *fido*  
  Fido  
  4.3.10

- **arc 2:** $N/V$ from *fido* to *dig*  
  dug  
  4.3.11

- **arc 3:** $V \setminus N$ from *dig* to *bone*  
  the_bone  
  4.3.12

- **arc 4:** $N \setminus V$ from *bone* to *dig*  
  $up_-$  
  4.3.13

The navigation resembles that of a content with a simple transitive verb. Realizing the discontinuous bare preposition is handled by lexverb alone.

The sequence of explicit speak mode operations begins with $V/N$:

#### 4.3.10 Navigating with $V/N$ from *dig* to *fido* (arc 1)

```
<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V/N$ (↓)</td>
<td></td>
</tr>
<tr>
<td>verb: $\alpha$</td>
<td>sur: lexnoun(β)</td>
</tr>
<tr>
<td>arg: $\beta \ X$</td>
<td>noun: β</td>
</tr>
<tr>
<td>prn: K</td>
<td>fnc: α</td>
</tr>
<tr>
<td></td>
<td>prn: K</td>
</tr>
<tr>
<td>↑</td>
<td>#-mark $\beta$ in the arg slot of proplet $\alpha$</td>
</tr>
<tr>
<td>sur: dig</td>
<td>sur: Fido</td>
</tr>
<tr>
<td>cat: #n &quot;n' #a' #bp' decl</td>
<td>noun: (dog x)</td>
</tr>
<tr>
<td>sem: $up$ past</td>
<td>cat: snp</td>
</tr>
<tr>
<td>arg: (dog x) bone</td>
<td>sem: nm m</td>
</tr>
<tr>
<td>mdr:</td>
<td>fnc: dig</td>
</tr>
<tr>
<td>nc:</td>
<td>mdr:</td>
</tr>
<tr>
<td>pc:</td>
<td>nc:</td>
</tr>
<tr>
<td>prn: 21</td>
<td>pc:</td>
</tr>
<tr>
<td></td>
<td>prn: 21</td>
</tr>
</tbody>
</table>
```
4.3.11 Navigating with N/V from *fido* back to *dig* (arc 2)

\[ \text{N/V (2)} \]

\[
\begin{array}{l}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K \\
\end{array}
\Rightarrow
\begin{array}{l}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{arg: } #\beta X \\
\text{prn: } K \\
\end{array}
\]

#-mark \( \alpha \) in the fnc slot of proplet \( \beta \)

\( Z \) is NIL, or elementary and #-marked

\[
\begin{array}{l}
\text{sur: } \text{fido} \\
\text{noun: } \text{(dog x)} \\
\text{cat: } \text{snp} \\
\text{sem: } \text{nm f} \\
\text{fnc: } \text{dig} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 21 \\
\end{array}
\]

\[
\begin{array}{l}
\text{sur: } \text{dig} \\
\text{cat: } \#u' \#a' \#bp' \text{ decl} \\
\text{sem: } \text{up past} \\
\text{arg: } #\text{(dog x) bone} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 21 \\
\end{array}
\]

4.3.12 Navigating with V\_N from *dig* to *bone* (arc 3)

\[ \text{V\_N (3)} \]

\[
\begin{array}{l}
\text{verb: } \alpha \\
\text{arg: } #X \beta Y \\
\text{prn: } K \\
\end{array}
\Rightarrow
\begin{array}{l}
\text{sur: } \text{lexnoun}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K \\
\end{array}
\]

#-mark \( \beta \) in the arg slot of proplet \( \alpha \)

\[
\begin{array}{l}
\text{sur: } \text{the_bone} \\
\text{noun: } \text{bone} \\
\text{cat: } \text{snp} \\
\text{sem: } \text{def sg} \\
\text{fnc: } #\text{dig} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 21 \\
\end{array}
\]

\[
\begin{array}{l}
\text{sur: } \text{up\_verb} \\
\text{cat: } \text{dig} \\
\text{sem: } \text{up past} \\
\text{arg: } #\text{(dog x) bone} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 21 \\
\end{array}
\]

4.3.13 Navigating with N\_V from *bone* back to *dig* (arc 4)

\[ \text{N\_V (4)} \]

\[
\begin{array}{l}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K \\
\end{array}
\Rightarrow
\begin{array}{l}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{arg: } #X \#\beta Y \\
\text{prn: } K \\
\end{array}
\]

\( Z \) is #-marked or NIL

\[
\begin{array}{l}
\text{sur: } \text{bone} \\
\text{cat: } \text{snp} \\
\text{sem: } \text{def sg} \\
\text{fnc: } #\text{dig} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 21 \\
\end{array}
\]

\[
\begin{array}{l}
\text{sur: } \text{up\_verb} \\
\text{cat: } \text{dig} \\
\text{sem: } \text{up past} \\
\text{arg: } #\text{(dog x) bone} \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 21 \\
\end{array}
\]
The \texttt{sem} value \textit{up} and the \texttt{cat} value \texttt{decl} are used by \texttt{lexverb} to realize \textit{up}.

Extending DBS.14 to an example class with a verb taking a discontinuous bare preposition required the definition of the additional hear mode operation \texttt{PRD×BP} (4.14). The hear mode derivation used five operation applications, and the speak mode derivation used four:

4.3.14 Comparing Hear and Speak Mode Surface Segmentation

hear mode:

\begin{verbatim}
1 2 3 4 5
Fido × dug × the ∪ bone ∪ up ∪ .
\end{verbatim}

speak mode:

\begin{verbatim}
1 2 3 4
V/N Fido N/V dug V\backslash N the_bone N\backslash V up_.
\end{verbatim}

The hear and the speak mode derivations run straight through.

4.4 DBS.16: Verb Taking an Infinitival Object

Verbs which take an infinitive as one of their arguments resemble those which take an obligatory prepositional phrase (Sect. 4.1), an adnominal (Sect. 4.2), or a bare preposition (Sect. 4.3) in that they form rather restricted selectional constellation classes (CLaTR Sect. 15.5). For example, the verbs taking a noun or an infinitive as object are restricted to \{\textit{begin, can afford, choose, decide, expect, forget, learn, like, manage, need, offer, plan, prepare, refuse, start, try, want}\}, a restriction which is formalized as a condition in \texttt{PRD×INF} (4.4.4).

The following content shows the verb \textit{try} taking the noun \textit{John} as subject and the infinitive to \textit{read a book} as object:

4.4.1 Content of the Surface \textit{John tried to read a book}.

\begin{verbatim}
[1sur: john
noun: (person x)
cat: snp
sem: nm m
dfn: try
mdr:
cnc:
pc:
prn: 24]

[2sur: verb: try
cat: #n #a' decl
sem: past
dfn: (person x)
mdr:
cnc:
pc:
prn: 24]

[3sur: verb: read
cat: #n-s3 #a' inf
sem: (person x)
mdr:
cnc:
pc:
prn: 24]

[4sur: noun: book
cat: snp
sem: indef sg
dfn: read
mdr:
cnc:
pc:
prn: 24]
\end{verbatim}

There is only one \texttt{prn} value, indicating an intrapropositional construction. The infinitival object is represented by the value \textit{read} in the second \texttt{arg} slot of \textit{try}. The role of \textit{John} as the implicit subject of the infinitive (“subject control”) is indicated by the value \textit{(person x)} in the first \texttt{arg} slot of \textit{read}.

Consider the graph of the time-linear hear mode derivation:
### 4.4.2 Graphical Hear Mode Derivation of the Content

#### 4.4.1

unanalyzed surface

<table>
<thead>
<tr>
<th>John</th>
<th>tried</th>
<th>to</th>
<th>read</th>
<th>a</th>
<th>book</th>
</tr>
</thead>
</table>

**automatic word form recognition**

```
1. noun: John  
   cat: sn  
   fnc: try  
   arg: [person x]  
   prn: 24
```

```
2. noun: John  
   cat: sn  
   fnc: try  
   arg: [person x]  
   prn: 24
```

```
3. noun: John  
   cat: sn  
   fnc: try  
   arg: [person x]  
   prn: 24
```

```
4. noun: John  
   cat: sn  
   fnc: try  
   arg: [person x]  
   prn: 24
```

```
5. noun: John  
   cat: sn  
   fnc: try  
   arg: [person x]  
   prn: 24
```

```
6. noun: John  
   cat: sn  
   fnc: try  
   arg: [person x]  
   prn: 24
```

**syntactic–semantic parsing**

| sur: John  
| noun: (person x)  
| cat: sn  
| sem: nm m  
| fnc: try  
| pm: 24  
| sur: to  
| arg: [person x]  
| pm: 24  
| sur: read  
| arg: [person x]  
| pm: 24  
| sur: a  
| pm: 24  
| sur: book  
| pm: 24  

result
The first hear mode operation to apply is standard SBJ×PRD:

4.4.3 CROSS-COPYING John AND tried WITH SBJ×PRD (line 1)

\[
\text{SBJ×PRD} \quad (\text{#1})
\]

\[
\begin{array}{c|c|c|c|c|c}
\text{pattern} & \text{level} & \text{content} & \text{level} & \text{content} & \text{level} \\
\hline
\text{noun: } \alpha & \text{cat: NP} & \text{verb: } \beta & \text{cat: NP' X v} & \text{noun: } \alpha & \text{cat: NP} \\
\text{fnc: } K & \text{arg: } \alpha & \text{fnc: } \beta & \text{arg: } \alpha & \\
\text{prn: } K & \text{prn: } K & \text{prn: } K & \text{prn: } K & \\
\end{array}
\]

Agreement conditions as in 2.1.2.

\[
\begin{array}{c|c|c|c|c|c}
\text{sur: John} & \text{cat: snp} & \text{cat: n' a' v} & \text{sem: past} & \text{nc: } (\text{person x}) & \text{pc: } 24 \\
\text{cat: #nom} & \text{arg: } v_1 & \text{fnc: try} & \text{mnr: } \text{pc: } 24 & \\
\text{sem: } \text{nm m} & \text{arg: } \alpha & \text{nc: } & \text{prn: } 24 & \\
\text{fnc: } & \text{prn: } 24 & \text{pc: } & \text{prn: } 24 & \\
\text{mnr: } & \text{prn: } 24 & \text{prn: } 24 & \text{prn: } 24 & \\
\end{array}
\]

The valency position of the direct object is \( \alpha' \). In verbs of the \textit{try} class (CLaTR 15.5.2), it may be canceled by a noun or by the conjunction \textit{to} of an infinitive.

The new operation PRD×INF\textit{try} connects \textit{try} and \textit{to} by means of cross-copying. The lexical analysis of \textit{to} contributes the core attribute \textit{verb} and the continuation attributes \textit{arg} and \textit{fnc}. The latter is normally reserved for nouns, but used here for the \textit{V/V} connection between the finite and the non-finite verb.

The \textit{arg} attribute is used for coding the subject of, e.g. \textit{try}, as the implicit subject of, e.g. \textit{read}. Called “subject control,” it is characteristic of the \textit{try}-class, listed below as the variable restriction of \( \alpha \) in PRD×INF\textit{try}:

4.4.4 CROSS-COPYING \textit{try} AND \textit{to} WITH PRD×INF\textit{try} (line 2)

\[
\text{PRD×INF\textit{try}} \quad (\text{#1})
\]

\[
\begin{array}{c|c|c|c|c|c}
\text{pattern} & \text{level} & \text{content} & \text{level} & \text{content} & \text{level} \\
\hline
\text{verb: } \alpha & \text{cat: #nom' a' v} & \text{verb: } v_1 & \text{cat: inf} & \text{verb: } \alpha & \text{cat: inf} \\
\text{arg: } \beta & \text{fnc: } & \text{arg: } \beta & \text{fnc: } & \text{arg: } \beta & \\
\text{prn: } K & \text{prn: } K & \text{prn: } K & \text{prn: } K & \\
\end{array}
\]

\( \alpha \in \{ \text{begin, can afford, choose, decide, expect, forget, learn, like, manage, need, offer, plan, prepare, refuse, start, try, want} \} \)

\[
\begin{array}{c|c|c|c|c|c}
\text{sur: } \text{try} & \text{cat: n' a' v} & \text{cat: inf} & \text{cat: #nom' a' v} & \text{cat: inf} & \text{sur: } \text{try} \\
\text{sem: past} & \text{sem: past} & \text{arg: } (\text{person x}) & \text{arg: } (\text{person x}) & \text{arg: } (\text{person x}) & \\
\text{arg: } (\text{person x}) & \text{mnr: } \text{mnr: } \text{arg: } (\text{person x}) & \text{nc: } \text{nc: } \text{nc: } & \text{pc: } \text{pc: } \text{pc: } & \text{prn: } \text{prn: } \text{prn: } \\
\text{fnc: } & \text{fnc: } & \text{fnc: } & \text{fnc: } & \text{fnc: } & \\
\text{mnr: } & \text{mnr: } & \text{mnr: } & \text{mnr: } & \text{mnr: } & \\
\text{nc: } & \text{nc: } & \text{nc: } & \text{nc: } & \text{nc: } & \\
\text{pc: } & \text{pc: } & \text{pc: } & \text{pc: } & \text{pc: } & \\
\text{prn: } 24 & \text{prn: } 24 & \text{prn: } 24 & \text{prn: } 24 & \text{prn: } 24 & \\
\end{array}
\]
The resulting \( v_1 \) proplet contributes the \texttt{cat} value \texttt{inf} and the implicit subject \texttt{(person x)} when the verb of the infinitive is absorbed next:

### 4.4.5 Absorbing \texttt{read} into \( v_1 \) with \texttt{INFV} (line 3)

In accordance with surface compositionality, the form \texttt{read} is analyzed in the input with the standard lexical feature \texttt{[cat: n-s3' \ a'] inf} for the unmarked present tense (NLC A.5.1). To satisfy the variable \#X\(^5\) of the following operation \texttt{PRD \times OBJ}, the implicit nominative valency position \#n-s3' is left in place.

Standard \texttt{PRD \times OBJ} is activated by the next word \texttt{a(n)}, which finds the output of \texttt{4.4.5} at the now front matching its first input pattern and applies:

### 4.4.6 Cross-copying \texttt{read} and \texttt{a(n)} with \texttt{PRD \times OBJ} (line 4)

---

\(^5\) As specified in 1.6.5(2), the underline of a variable requires at least one matching value.
If the verb of the infinitive where three-place, as in *to give Mary a kiss*, the operation would apply once more, based on the feature [cat: #X’ N’ Y γ] of the first input pattern. Here, however, *read* provides only one oblique valency. The direct object of the infinitive is completed with DET∪CN:

### 4.4.7 Absorbing *book* into n_1 with DET∪CN (line 5)

The hear mode derivation concludes with adding the period:

### 4.4.8 Absorbing *.* into try with S∪IP (line 6)

The apparent distance between the period proplet and the top verb in line 6 of the graph 4.4.2 is of no relevance: whether or not a proplet at the current now front matches the first input pattern of a hear mode operation is determined solely by its core attribute and other, similar properties of its feature structure, and not by distance in the linear surface.

We continue with point 4 of 1.5.1, i.e. the canonical DBS graph analysis for the speak mode. Unlike 4.1.10 and 4.2.8 it uses a depth first arc numbering:
4.4.9 **Graph Analysis Underlying Production of**

(i) **SRG (semantic relations graph)**

(ii) **signature**

(iii) **NAG (numbered arcs graph)**

(iv) **surface realization**

The V\V line characterizes the infinitive as a grammatical object, while the V\N line characterizes the infinitive verb as transitive.

Following standard procedure, we continue with point 5 on the to-do list 1.5.1, i.e. the list of speak mode operation names with surface realization:

4.4.10 **Sequence of Operation Names and Surface Realizations**

- **arc 1:** V/N from *try* to *john*  
  John
  
- **arc 2:** N/V from *john* to *try*  
  tried

- **arc 3:** V\INF\Try from *try* to *read*  
  to_read

- **arc 4:** INF\N from *read* to *book*  
  a_book

- **arc 5:** N\INF from *book* to *read*  

- **arc 6:** INF\V from *read* to *try*  

The next item of 1.5.1 is point 6, i.e. the sequence of explicit speak mode applications. The navigation begins with standard V/N traversing arc 1:

4.4.11 **Navigating with V/N from *try* to *john* (arc 1)**

*Pattern level*

- **verb:** α
- **arg:** β X
- **prn:** K

*Content level*

| sur: John |
| noun: (person x) |
| cat: snp |
| sem: nm m |
| fnc: try |
| mdr: |
| nc: |
| pc: |
| prn: 24 |

#-mark β in the arg slot of proplet α
Using the continuation feature [fnc: try] of the goal proplet, the navigation returns with N/\(V\) to the top verb and realizes tried:

4.4.12 Navigating with N/\(V\) from John back to try (arc 2)

\[
\text{N/\(V\) (5)}
\begin{align*}
\text{pattern level} & : \\
\text{noun: } & \beta \\
\text{fnc: } & \alpha \\
\text{mdir: } & Z \\
\text{prn: } & K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: lexverb(\(\hat{\alpha}\))} \\
\text{verb: } & \alpha \\
\text{arg: } & \#\beta \ X \\
\text{prn: } & K
\end{align*}
\text{#-mark } \alpha \text{ in the fnc slot of proplet } \beta

Z is NIL, or elementary and #-marked

\[
\text{content level} \\
\text{sur: } & \text{John} \\
\text{noun: } & (\text{person } x) \\
\text{cat: } & \text{snp} \\
\text{sem: } & \text{nm m} \\
\text{fnc: } & \text{try} \\
\text{mdir: } & \text{nc} \\
\text{pc: } & \text{24} \\
\text{prn: } & K
\]

The verbal arg value \(\text{read}\) of try matches the goal pattern of the new operation \(V \setminus \text{INF}_{\text{try}}\), which – like \(\text{PRD} \times \text{INF}_{\text{try}}\) (4.4.4) – is limited to verbs of the try-class (CLaTR 15.5.2), in contradistinction to other classes of verbs taking an infinitive as object, such as the appear-class (CLaTR 15.6.3), the promise-class (CLaTR 15.6.5), and the persuade-class (CLaTR 15.6.7):

4.4.13 Navigating with \(V \setminus \text{INF}_{\text{try}}\) from try to read (arc 3)

\[
\text{V \setminus \text{INF}_{\text{try}} (33)}
\begin{align*}
\text{pattern level} & : \\
\text{verb: } & \alpha \\
\text{arg: } & \#\gamma \ \beta \\
\text{prn: } & K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: lexverb(\(\hat{\beta}\))} \\
\text{verb: } & \beta \\
\text{cat: } & \#NP \ #X \ \text{inf} \\
\text{fnc: } & \#\alpha \\
\text{arg: } & \#\gamma \ Y \\
\text{prn: } & K
\end{align*}
\text{#-mark } \beta \text{ in the arg slot of proplet } \alpha

\alpha \in \{\text{begin, can afford, choose, decide, expect, forget, learn, like, manage, need, offer, plan,}
\text{ prepare, refuse, start, try, want}\}

\[
\text{content level} \\
\text{sur: } & \text{to\_read} \\
\text{verb: } & \text{read} \\
\text{cat: } & \#n-x3' \ #a' \ \text{inf} \\
\text{sem: } & \text{fnc: } \#\text{try} \\
\text{arg: } & \#(\text{person } x) \ \text{book} \\
\text{nc: } & \text{pc:} \\
\text{prn: } & \text{24}
\]
The variable $\gamma$ in the initial arg slot of $\beta$ indicates “subject control” of the try-class. The additional arg feature of read originated lexically from the intrapropositional conjunction to in the hear mode (4.4.4).

Even though the valency positions $\#n'$ of try and $\#n\cdot s3'$ of read are different, they relate to the same filler (person x). In the present case, valency positions for the same filler may differ because the infinitive verb is fixed to the subject valency $n\cdot s3'$ (and the sem value pres), regardless of what the underlying subject stands for. For example, John wants to eat a cookie is grammatical, even though (*John) eat a cookie fails to agree.

The traversal of the infinitive continues with the direct object, using the continuation value book in the second arg slot of read:

4.4.14 Navigating with INF\N from read to book (arc 4)

$$\begin{align*}
\text{pattern level} & : \text{verb: } \alpha \\
& \quad \text{arg: } \#X \beta Y \\
& \quad \text{prn: } K \\
\Rightarrow & : \text{sur: } \text{lexnoun}(\hat{\beta}) \\
& \quad \text{noun: } \beta \\
& \quad \text{fnc: } \alpha \\
& \quad \text{prn: } K \\
\text{content level} & : \text{sur: a_book} \\
& \quad \text{verb: read} \\
& \quad \text{cat: } \#n\cdot s3' \#a' \text{ inf} \\
& \quad \text{sem: } \\
& \quad \text{fnc: #try} \\
& \quad \text{arg: } \text{#(person x) book} \\
& \quad \text{nc: } \\
& \quad \text{pc: } \text{prn: 24} \\
\end{align*}$$

After realizing the surface a_book from the book proplet, the navigation returns empty to read:

4.4.15 Navigating with N\INF from book back to read (arc 5)

$$\begin{align*}
\text{pattern level} & : \text{noun: } \beta \\
& \quad \text{fnc: } \#\alpha \\
& \quad \text{prn: } K \\
\Rightarrow & : \text{sur: } \text{lexverb}(\hat{\alpha}) \\
& \quad \text{verb: } \alpha \\
& \quad \text{arg: } \#X \beta \ Y \\
& \quad \text{prn: } K \\
\text{content level} & : \text{sur: verb: read} \\
& \quad \text{cat: } \#n\cdot s3' \#a' \text{ inf} \\
& \quad \text{sem: } \\
& \quad \text{fnc: #try} \\
& \quad \text{arg: } \text{#(person x) book} \\
& \quad \text{nc: } \\
& \quad \text{pc: } \text{prn: 24} \\
\end{align*}$$
Finally, the navigation returns to try and realizes the surface . from the decl value in the cat slot of the goal proplet.

### 4.4.16 Navigating with INF\V from read back to try

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: (\beta)</td>
<td>sur: lexverb((\gamma))</td>
</tr>
<tr>
<td>cat: #X inf</td>
<td>verb: (\alpha)</td>
</tr>
<tr>
<td>arg: #Y</td>
<td>arg: (\beta) X</td>
</tr>
<tr>
<td>prn: K</td>
<td>prn: K</td>
</tr>
</tbody>
</table>

\(\gamma \in \{\text{that, when, where, how, why}\}\)

\[ \Rightarrow \]

| sur: read | sur: • |
| verb: try | verb: try |
| cat: #n-s3 #a’ decl | cat: #n’ #a’ decl |
| sem: past | sem: past |
| arg: #(person x) #read | arg: #(person x) #read |
| mdr: | mdr: |
| nc: | nc: |
| pc: | pc: |
| prn: 24 | prn: 24 |

In summary, once the verb of an infinitival object has been added by fusing it with to (4.4.5), the continuation of the infinitival phrase is free: the infinitive phrase may range from a bare one-place verb, as in John tried to sleep to constructions of arbitrary grammatical complexity.

Let us follow standard procedure and conclude the section with point 7 on the to-do list [1.5.1]. Extending DBS.15 to an example class with an infinitival object required the definition of the two hear mode operations PRD\(\times\)INF\try (h11) and INF\(\cup\)V (h17), and the four speak mode operations INF\N (s31), N\INF (s32), INF\V (s33), and INF\V (s34).

The hear mode and the speak mode derivations used six operation applications each:

### 4.4.17 Comparing Hear and Speak Mode Surface Segmentation

**Hear mode:**

1 2 3 4 5 6

John \(\times\) tried \(\times\) to \(\cup\) read \(\times\) a \(\cup\) book \(\cup\) .

**Speak mode:**

1 2 3 4 5 6

V/N John N/V tried V\INF\try to_read INF\N a book N\INF INF\V .

The hear mode derivation runs straight through, the speak mode derivation has one empty traversal.
4.5 DBS.17: Copula be Taking a Noun

The English auxiliary be may be used in complex verb constructions (Sects. 2.2, 3.1) or as a copula. In either case, be is like an almost empty verb shell which specifies tense and mood, and requires subject agreement but lacks a more substantial content beyond some general notion of equation.

As an auxiliary, be takes a non-finite form of a main verb for more differentiated content, as in was running. As a copula, be takes a second noun, as in is a doctor, or an adnominal, as in is hungry.

Regarding the grammatical role of a second noun, English syntax provides no clear criterion as to whether it is a nominative or an oblique. For example, when looking at an old foto, one may point at a person and say “This is me.” On the phone, in contrast, one may reply to the request of speaking to a certain person with “This is I.” In consequence we use the operation PRD × OBJ for connecting the second noun; pronominal nominatives in post-copula position are accommodated by the variable restrictions of 2.3.4.

As an example consider the following content with be taking a subject noun and a second noun, which has been called “predicative” in English grammar:

4.5.1 CONTENT OF THE SURFACE Julia is a doctor.

The core value of the proplet representing is is the substitution variable \( v_1 \). The equation quality is coded by the valency position be' in the cat slot. As shown by the following hear mode derivation in canonical DBS graph format, the valency position be' is filled by the second noun doctor in lines 2 and 3:

4.5.2 GRAPHICAL HEAR MODE DERIVATION OF THE CONTENT 4.5.1

unanalyzed surface

| sur: Julia  | is    | a   | doctor
| noun: (person x) | verb: v_1 | arg: fnc: v_2 | decl: cat: v_2
| cat: sn | cat: #ns3' #be' | arg: fnc: | cat: sn |
| sem: nm f | sem: pres | sem: indef sg | sem: sg |
| prn: 22 | prn: 22 | prn: 22 | prn: 22 |

automatic word form recognition

| sur: Julia  | sur: is  | sur: a   | sur: doctor |
| noun: (person x) | verb: v_1 | noun: n_1 | noun: doctor |
| cat: sn | cat: #ns3' be' | cat: sn' np | cat: sn |
| sem: nm f | sem: pres | sem: indef sg | sem: sg |
| prn: pm: | prn: pm: | prn: pm: | prn: pm: |
4.5 DBS.17: Copula be Taking a Noun

The auxiliary character of the copula is represented by the substitution variable \( v_1 \) serving as the core value. The kind of copula is coded by the valency position \( \text{be}' \), distinguishing it from \( \text{hv}' \) and \( \text{do}' \) (CLaTR 6.6.8). The core value of \( \text{doctor} \) is copied into the second \( \text{arg} \) slot, i.e. the standard object position.

The first operation to apply is \( \text{SBJ} \times \text{PRD} (2.1.2) \). It does not require any adjustment because the variable \( X \) in the feature \( [\text{cat: NP}' X \text{ v}] \) matches any oblique argument value, including \( \text{be}' \):

---

6 In addition, there are the auxiliaries \( \text{have} \) and \( \text{do} \) in English (FoCL Sect. 17.4), as well as the modals \( \text{can}, \text{could}, \text{may}, \text{might}, \text{must}, \text{ought}, \text{shall}, \text{should}, \text{will}, \) and \( \text{would} \). An example of using \( \text{have} \) as an auxiliary is \( \text{John has bought a car} \); an example of using \( \text{have} \) as a copula is \( \text{John has a car} \) (general relation of ownership). An example of using \( \text{do} \) as an auxiliary is emphatic \( \text{John does like Mary} \); an example of \( \text{do} \) as a copula is \( \text{Mary does the boogie-woogie} \).

7 In German, the second noun is morphologically a clear nominative, called \( \text{Gleichsetzungsnominativ} \) in the grammar literature. For example, it is \( \text{Das bin ich} \), but not *\( \text{Das bin mich} \), *\( \text{Das bin mir} \), or *\( \text{Das ist mich} \), which are all severely ungrammatical. Yet there is \( \text{Wie ist Dir?} \), which is archaic but grammatical and documented (Brömel 1794, p. 15) – in contradistinction...
4.5.3 Cross-copying Julia and is with SBJ×PRD (line 1)

The agreement condition of SBJ×PRD works in the usual way.

The canceling of the ns3' value in the v_/I proplet enables the PRD×OBJ operation to apply next. The agreement between the valency position be' and the associated filler snp is accommodated by the variable restrictions of 2.3.4.

4.5.4 Cross-copying v_/I and a with PRD×OBJ (line 2)

With the second valency position be' in the cat slot of v_/I canceled by the determiner of a second nominal argument (and not by an adnominal modifier like hungry as in 4.6.1), it remains to complete the noun.

Automatic word form recognition provides the common noun doctor as the next word and standard DET\cup CN applies as follows:

to "Wie bist Du?, which uses a Gleichsetzungsnominativ but is not employed in German, despite the English analog. However, Where am I?/Wo bin ich?, Where is she?/Wo ist sie?, etc., work alike.
4.5.5 **ABSORBING** doctor **into** n_\textsubscript{1} **with** DET\textcup CN (line 3)

\[
\begin{array}{c}
\text{pattern level} \\
\text{DET} \cup \text{CN (h17)} \\
\begin{array}{|c|}
\hline
\text{noun: N}_n \\
\text{cat: X CN' NP} \\
\text{sem: Y} \\
\text{prn: K} \\
\hline
\end{array} \\
\begin{array}{|c|}
\hline
\text{noun: } \alpha \\
\text{cat: CN} \\
\text{sem: Z} \\
\text{prn: } K \\
\hline
\end{array} \\
\Rightarrow
\begin{array}{|c|}
\hline
\text{noun: } \alpha \\
\text{cat: X NP} \\
\text{sem: } Y \ Z \\
\text{prn: } K \\
\hline
\end{array}
\end{array}
\]

Agreement conditions as in 2.2.4.

\[
\begin{array}{c}
\text{content level} \\
\text{sur: } \text{n}_1 \\
\text{cat: sn' snp} \\
\text{sem: indef} \\
\text{fnc: } \text{v}_1 \\
\text{mdr: } \text{nc:} \\
\text{pc: } \text{prn: 22} \\
\hline
\text{sur: } \text{doctor} \\
\text{cat: sn} \\
\text{sem: sg} \\
\text{fnc: } \text{v}_1 \\
\text{mdr: } \text{nc:} \\
\text{pc: } \text{prn: 22} \\
\hline
\text{sur: } \text{doctor} \\
\text{cat: snp} \\
\text{sem: indef sg} \\
\text{fnc: } \text{v}_1 \\
\text{mdr: } \text{nc:} \\
\text{pc: } \text{prn: 22} \\
\hline
\end{array}
\]

The hear mode derivation concludes with standard S\textcup IP adding the punctuation, here the period : .

4.5.6 **ABSORBING** \textbullet \textsubscript{.} **into** v_\textsubscript{1} **with** S\textcup IP (line 7)

\[
\begin{array}{c}
\text{pattern level} \\
\text{S\textcup IP (h15)} \\
\begin{array}{|c|}
\hline
\text{verb: } \beta \\
\text{cat: } \text{#X' VT} \\
\text{prn: } K \\
\hline
\end{array} \\
\begin{array}{|c|}
\hline
\text{verb: } \text{V}_n \\
\text{cat: } \text{VT' SM} \\
\hline
\end{array} \\
\Rightarrow
\begin{array}{|c|}
\hline
\text{verb: } \beta \\
\text{cat: } \text{#X' SM ip+} \\
\hline
\end{array}
\end{array}
\]

Agreement conditions as in 2.1.3.

\[
\begin{array}{c}
\text{content level} \\
\text{sur: } \text{v}_1 \\
\text{cat: } \text{#ns3' #be' v} \\
\text{sem: pres} \\
\text{arg: (person x) doctor} \\
\text{mdr: } \text{nc:} \\
\text{pc: } \text{prn: 22} \\
\hline
\text{sur: } \text{•} \\
\text{cat: } \text{v' decl} \\
\text{prn: } \\
\hline
\text{sur: } \text{v}_1 \\
\text{cat: } \text{#ns3' #be' decl +ip} \\
\text{sem: pres} \\
\text{arg: (person x) doctor} \\
\text{mdr: } \text{nc:} \\
\text{pc: } \text{prn: 22} \\
\hline
\end{array}
\]

In summary, the case role of post-copula nouns is left undecided because (i) determiner-noun combinations and names have no concrete morphological case marking in English and (ii) those which do, i.e. pronouns like she vs. her, provide conflicting evidence. Therefore, we rely on English word order and treat non-initial arguments in declarative sentences generally as grammatical objects, even if they are pronouns with a case-marking for nominative.

Following standard procedure, we continue with point 4 on the DBS to-do list \[1.5.1\] i.e. the canonical graph analysis for the speak mode.
4.5.7 Graph Analysis Underlying Production of 4.5.1

(i) SRG (semantic relations graph)

\[ \begin{array}{c}
\text{be} \\
\text{julia} \quad \text{doctor}
\end{array} \]

(ii) signature

\[ \begin{array}{c}
\text{V} \\
\text{N} \\
\text{N}
\end{array} \]

(iii) NAG (numbered arcs graph)

\[ \begin{array}{c}
1 \\
2 \\
3 \\
4
\end{array} \]

(iv) surface realization

\[ \begin{array}{c}
\text{Julia} \\
\text{is} \\
\text{a_doctor}
\end{array} \]

The graphs (i)–(iii) show the copula with the subject and a second noun. The semantic relations in the third line of the (iv) surface realization resembles those of a simple transitive verb structure such as 2.3.6.

The next item on the to-do list 1.5.1 is point 5, i.e. the list of speak mode operation names with surface realizations:

4.5.8 Sequence of Operation Names and Surface Realizations

arc 1: V/N from be to julia Julia 4.5.9
arc 2: N/V from julia to be is 4.5.10
arc 3: V\N from is to doctor a_doctor 4.5.11
arc 4: N\V from doctor to be 4.5.12

The next item on the to-do list 1.5.1 is point 6, i.e. the sequence of explicit speak mode operations. The first operation to apply is V/N.

4.5.9 Navigating with V/N from v_1 to julia (arc 1)

\[ \begin{array}{c}
\text{sur: lexnoun(} \beta \text{)} \\
\text{noum: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K
\end{array} \]

\[ \begin{array}{c}
\text{sur: Julia} \\
\text{noum: (person x)} \\
\text{cat: snp} \\
\text{sem: nm f} \\
\text{fnc: v_1} \\
\text{mdr: } \text{nc: } \text{pc: } \text{prn: 22}
\end{array} \]

There is no need to adjust V/N for this application because the variable \( \alpha \) is not defined to exclude a substitution variable like v_1 as a possible value to be bound to.
The same holds for N/V, which applies next:

4.5.10 **Navigating with N/V from julia back to v_1** (arc 2)

Using the sole continuation feature \([\text{fnc: } v_1]\), the navigation returns to the copula. Using the cat value \(\text{be}'\) and the sem value \(\text{pres}\) of the goal proplet, \(\text{lexverb}\) realizes the surface \(\text{is}\).

With the first arg value \((\text{person x})\) #-marked, the sole continuation value for the navigation to proceed from the copula is the second arg value \(\text{doctor}\) using V\N:

4.5.11 **Navigating with V\N from v_1 to doctor** (arc 3)

Lexnoun realizes the surface \(\text{a_doctor}\) from the sem values \(\text{indef sg}\) and the core value \(\text{doctor}\).

Finally, the navigation returns to the copula with N\V:
4.5.12 Navigating with N\V from doctor back to v_1 (arc 4)

Using the cat value decl of the goal proplet, lexverb realizes the surface ●.

We follow standard procedure and conclude the section with point 7 on the to-do list [4.5.1] Applying DBS.16 to an example class with the copula be taking a noun did not require the definition of an additional operation. The hear and the speak mode derivations used four operation applications each, but the surface segmentation is different:

4.5.13 Comparing hear and speak mode surface segmentation

hear mode: 1 2 3 4
Julia × is × a ∪ doctor × .

speak mode: 1 2 3 4
V\N Julia N\V is V\N a_doctor N\V .

The hear and the speak mode derivations run straight through.

4.6 DBS.18: Copula be Taking an Adnominal

The previous section analyzed the expression Julia is a doctor., i.e. the copula be with a noun as the second argument. We turn now to analyzing the expression Fido is hungry, i.e. the copula be with an adnominal as the second argument. In both copula constructions, the second argument is obligatory in the sense that they would be incomplete without it.

As in [4.1.1] and [4.2.2] the obligatory nature of the modifier as an argument is expressed by the cat value adn of hungry canceling the valency position be' of the copula in addition to cross-copying the core values into the mdr and
4.6 DBS.18: Copula be Taking an Adnominal

mdd slots of the predicate and the object. Following standard procedure, we begin with point 1 of 1.5.1

4.6.1 CONTENT OF THE SURFACE Fido is hungry.

Let us follow standard procedure and continue with point 2 on the to-do list 1.5.1, i.e. the time-linear hear mode derivation in canonical DBS graph format:

4.6.2 GRAPHICAL HEAR MODE DERIVATION OF THE CONTENT 4.6.1

unanalyzed surface

Fido is hungry

automatic word form recognition

syntactic–semantic parsing

1. cross-copying

2. cross-copying

3. absorption
Following standard procedure, we continue with point 3 on the to-do list i.e. the complete sequence of explicit hear mode operation applications.

### 4.6.3 CROSS-COPYING Fido and is with SBJ×PRD (line 1)

At this point, the hear mode has derived Fido is. Next, automatic word form recognition provides the adn hungry. It is combined with the copula by the operation PRD×ADN (4.2.6):
The operation cancels the valency position $\text{be}'$ with the filler $\text{adn}$ and cross-copies the core values of the copula and $\text{hungry}$ into their respective $\text{mdr}$ and $\text{mdd}$ slots. The restriction of the variable $\gamma$ to $\text{mdr}'$ and $\text{be}'$ allows the use of $\text{PRD} \times \text{ADN}$ in 4.2.6 and 4.6.4; the restriction of $\text{ADN}$ to $\{\text{adn}, \text{adnv}\}$ in the case of $\text{be}'$ allows a phrasal modifier, as in Fido is in the kitchen.

The hear mode derivation concludes with standard $\text{S} \cup \text{IP}$:

4.6.5 **ABSORBING $\bullet$ INTO $v_1$ WITH $\text{S}\cup\text{IP}$ (line 3)**

In summary, the use of nouns vs. modifiers as arguments is parallel in a main verb and a copula: noun arguments are stored in the $\text{arg}$ slot (2.4.2, line 4; 4.5.2, line 2) while modifier arguments are stored in the $\text{mdr}$ slot (4.1.2, line 4; 4.2.2, line 4; 4.6.2, line 2). Furthermore, modifiers used as arguments cancel a valency position in the $\text{cat}$ slot of the predicate (4.1.2, line 4; 4.6.2, line 2), while optional modifiers do not (3.1.5, 3.1.6, 3.1.8, 3.1.9; 3.2.7).

The next point 4 on the to-do list 1.5.1 is the DBS graph structure of the speak mode, based on the content derived in the hear mode (4.6.2):

4.6.6 **GRAPH ANALYSIS UNDERLYING PRODUCTION OF 4.6.1**

Comparison with 4.5.7 for Julia is a doctor clearly shows the semantic difference between the two constructions.
Point 5 on the to-do list is the list of speak mode operation names and their surface realizations:

### 4.6.7 Sequence of Operation Names and Surface Realizations

- **arc 1:** $V/\searrow N$ from *be* to *fido*  
  *Fido*  
- **arc 2:** $N/\nearrow V$ from *fido* to *be*  
  *is*  
- **arc 3:** $V\downarrow A$ from *be* to *hungry*  
  *hungry*  
- **arc 4:** $A\uparrow V$ from *hungry* to *be*.

Following standard procedure, we continue with point 6 on the to-do list, i.e. the sequence of explicit speak mode operation applications:

#### 4.6.8 Navigating with $V/N$ from $v_1$ to *fido* (arc 1)

\[
\begin{align*}
\text{pattern level} & : \begin{cases}
\text{verb: } \alpha \\
\text{arg: } \beta X \\
\text{prn: } K
\end{cases} \\
\text{content level} & : \begin{cases}
\text{sur: } \text{lexnoun}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{nc: } \\
\text{pc: } \\
\text{prn: } 23
\end{cases}
\]

Even though the first input proplet differs from the one in by having a nonempty mdr slot, $V/N$ applies here all the same because its input pattern ignores the mdr feature. The next operation returns to the verb:

#### 4.6.9 Navigating with $N/V$ from *fido* back to $v_1$ (arc 2)

\[
\begin{align*}
\text{pattern level} & : \begin{cases}
\text{nou}\text{n: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K
\end{cases} \\
\text{content level} & : \begin{cases}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{arg: } \#\beta X \\
\text{prn: } K
\end{cases}
\]

$Z$ is NIL, or elementary and #-marked.
Having traversed the arcs 1 and 2 in 4.6.6 to realize *Fido* and *is*, the navigation continues with arcs 3 and 4 to realize *hungry* and *is*. For a speak mode navigation along existing semantic relations there is no need to distinguish between an optional modifier vs. a modifier argument. Therefore the operations $V \downarrow A$ and $A \uparrow V$ defined in 3.1.13 and 3.1.14 may be reused:

### 4.6.10 Navigating with $V \downarrow A$ from v₁ to *hungry* (arc 3)

\[
\begin{array}{c}
\text{pattern level} \\
\begin{array}{c}
\text{verb: } \alpha \\
\text{mdr: } #X \beta Y \\
\text{prn: } K \\
\end{array} \\
\uparrow \\
\begin{array}{c}
\text{sur: } \text{lexadj}(\hat{\beta}) \\
\text{adj: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K \\
\end{array}
\Rightarrow
\begin{array}{c}
\text{content level} \\
\begin{array}{c}
\text{sur: } \text{hungry} \\
\text{adj: } \text{hungry} \\
\text{cat: } \text{adn} \\
\text{sem: } \text{pad} \\
\text{mdd: } #v_1 \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \text{prn: } 23 \\
\end{array}
\end{array}
\end{array}
#-mark \beta in the mdr slot of proplet \alpha

With the following application of $A \uparrow V$ the navigation returns to the copula as the top verb, allowing realization of the period with lexverb:

### 4.6.11 Navigating with $A \uparrow V$ from *hungry* back to v₁ (arc 4)

\[
\begin{array}{c}
\text{pattern level} \\
\begin{array}{c}
\text{adj: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K \\
\end{array} \\
\uparrow \\
\begin{array}{c}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{mdr: } #X \beta Y \\
\text{prn: } K \\
\end{array}
\Rightarrow
\begin{array}{c}
\text{content level} \\
\begin{array}{c}
\text{sur: } * \\
\text{adj: } \text{hungry} \\
\text{cat: } \text{adn} \\
\text{sem: } \text{pad} \\
\text{mdd: } #v_1 \\
\text{mdr: } \\
\text{nc: } \\
\text{pc: } \text{prn: } 23 \\
\end{array}
\end{array}
\end{array}
\]

If the nc slot had an extrapropositional value, the navigation would continue from here to the next proposition.

Following standard procedure, we conclude this section with point 7 on the to-do list 1.5.1. Extending DBS.17 to an example class with the copula *be*
taking an adnominal did not require the definition of any additional operation. The hear mode derivation used three operation applications and the speak mode derivation used four:

4.6.12 Comparing hear and speak mode surface segmentation

hear mode:  
  1  2  3  Fido \times is \times hungry \cup .

speak mode:  1  2  3  4  V/N Fido N/V is V\downarrow A hungry A\uparrow V.

The hear and the speak mode derivations run straight through.
5. Unbounded Repetition

In linguistics, natural language has been claimed to be \textit{recursive} (Chomsky 1957; Hauser, Chomsky, and Fitch 2002). In computer science, a function is called recursive if it calls its input in its output, as in $x = x+1$, $\text{NP} \rightarrow \text{NP}$ $\text{S}$, or $\text{N} \rightarrow \text{A}$ $\text{N}$, resulting in an unlimited number of self-similar items, as in fractals.

Recently, the presence of recursion in natural language has been questioned, starting with Everett’s (2005) analysis of Pirahã, an indigenous language in Brazil. In the syntactic-semantic analysis of DBS, recursion in the sense of computer science is absent as well: grammatical constructions formerly presented as instances of substitution-based recursion are reanalyzed as the unbounded repetition of address-based continuations.

5.1 DBS.19: Unbounded Repetition of Prepositional Phrases

Prepositional phrases may be used not only for adnominal (3.2.1) and adverbial (3.2.2) modification, but also be repeated in unbounded coordination and modification relations (NLC Sects. 15.3, 15.4). Consider the following content with a repeating modification in adnominal use:

5.1.1 ADNOMINAL CONTENT OF THE SURFACE Fido ate the bone on the table under the tree in the garden...

The first prepositional phrase \textit{table} is connected to the grammatical object \textit{bone} by cross-copying the core values into the respective \textit{mdr} and \textit{mdd} slots. The second prepositional phrase \textit{tree}, in turn, is connected to \textit{table} by another
cross-copying of core values into the respective \( \text{mdr} \) and \( \text{mdd} \) slots, etc. The unboundedly repeating relation is modification because under the tree modifies table and in the garden modifies tree.

For reasons of space, the following time-linear hear mode derivation in DBS graph format omits the beginning of the example, which is shown in Section 3.2.3.

### 5.1.2 Graphical Hear Mode Derivation of the Content

<table>
<thead>
<tr>
<th>bone</th>
<th>on the table</th>
<th>under the tree</th>
<th>in the garden</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>sur: bone</td>
<td>sur: on</td>
<td>sur: table</td>
<td>sur: under the</td>
<td>sur: under the</td>
</tr>
<tr>
<td>cat: adnv</td>
<td>mtt: on</td>
<td>mtt: under</td>
<td>mtt: on</td>
<td>mtt: on</td>
</tr>
<tr>
<td>sem: def</td>
<td>mtt: def</td>
<td>mtt: def</td>
<td>mtt: def</td>
<td>mtt: def</td>
</tr>
<tr>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
</tr>
<tr>
<td>prn:</td>
<td>prn:</td>
<td>prn:</td>
<td>prn:</td>
<td>prn:</td>
</tr>
</tbody>
</table>

**Syntactic-semantic parsing**

1. cross-copying
2. absorption with simultaneous substitution
3. absorption with simultaneous substitution
4. cross-copying
5. absorption with simultaneous substitution
6. absorption with simultaneous substitution
7. cross-copying
8. absorption with simultaneous substitution
9. absorption with simultaneous substitution
Theoretically, the addition of prepositional phrases may be continued indefinitely. That this process does not involve recursion in any formal sense is shown by the cross-copyings in lines 4, 7, and 10, and by the following sequence of hear mode operations, specifically by the applications of $\text{N} \times \text{PREP}$ in 5.1.3, 5.1.6, and 5.1.9 which code inter-proplet relations by address, and not by an embedding which results from substitution.

5.1.3 CROSS-COPYING bone and on with $\text{N} \times \text{PREP}$ (line 4)

Combining bone and on provides the beginning of a prepositional modifier.

\[\text{N} \times \text{PREP}\]

\[
\begin{array}{c}
\text{pattern level} \\
\text{content level}
\end{array}
\]

---

1 The continuation in line 4 of 3.2.3 is adverbial $\text{PRD} \times \text{PREP}$, in contradistinction to the present example, which is continued with adnominal $\text{N} \times \text{PREP}$. 5.1.9.
The operation $\text{PREP} \cup \text{NP}$ (5.2.8) absorbs the determiner into the preposition (CLaTR 7.2.5):
5.1.6 CROSS-COPYING *table* AND *under* WITH $N \times PREP$ (line 7)

The second prepositional phrase is continued with $PREP \cup NP$ adding the determiner:

5.1.7 ABSORBING the INTO *n_4* WITH $PREP \cup NP$ (line 8)

The second input pattern would also accept an elementary noun, as in to you or in Paris. Here, however, the prepositional phrase must be completed with an application of $DET \cup CN$. The incrementation of the $N \_n$ substitution variable is seen best in the top line (automatic word form recognition) of 5.1.8.

5.1.8 ABSORBING *tree* INTO *n_5* WITH $DET \cup CN$ (line 9)

---

2 The automatic incrementation of the $N \_n$ variable is needed to keep track of different determiners, as in the German example *Der die das Kind fütternde Frau liebende Mann...*, which may be contrived but is grammatical.
The unbounded repetition of modification with prepositional phrases continues by adding the third preposition in to (under the) tree:

5.1.9 CROSS-COPYING tree AND in WITH N×PREP (line 10)

5.1.10 ABSORBING the INTO n_6 WITH PREP\_NP (line 11)
5.1.11 **ABSORBING garden into \( n_7 \) with \( \text{DET}_\cup \text{CN} \) (line 12)**

\[
\begin{align*}
\text{DET}_\cup \text{CN} \quad \text{(h37)} \\
\text{noun: } N_\alpha \\
\text{cat: } X \text{ CN' NP} \\
\text{sem: } Y \\
\text{prn: } \alpha \\
\end{align*}
\]

Agreement conditions as in 2.2.4.

\[
\begin{align*}
\text{sur: } n_7 \\
\text{cat: } \text{adnv} \text{ nn' np} \\
\text{sem: } \text{in} \text{ def} \\
\text{mdr: } \text{tree} \\
\text{nc: } \\
\text{pc: } 25 \\
\text{prn: } 25
\end{align*}
\]

The derivation may continue with another prepositional phrase or conclude with the period, depending on the hear mode input.

The next item on the to-do list 1.5.1 is the graph analysis of the speak mode:

5.1.12 **Graph analysis underlying production of 5.1.1**

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

\[
\begin{align*}
\text{eat} \\
\text{fido} & \quad \text{bone} \\
\text{table} & \quad \text{tree} \\
\text{garden}
\end{align*}
\]

\[
\begin{align*}
\text{eat} \\
\text{fido} & \quad \text{bone} \\
\text{table} & \quad \text{tree} \\
\text{garden}
\end{align*}
\]

The unbounded repetition of prepositional phrases, here two, is based on the local, binary \( N \mid N \) relation between the modifier and the modified, coded by address. Comparison with 3.6.10 shows the semantic difference between the intrapositional repetition of modification and of coordination.

Let us follow standard procedure and continue with point 5 on the DBS to-do list 1.5.1 i.e. the list of speak mode operation names with surface realizations.
5.1.13 **SEQUENCE OF OPERATION NAMES AND SURFACE REALIZATIONS**

- **arc 1:** \( V/N \) from *eat* to *fido*  
  \[ \text{Fido} \]  
  [5.1.14]

- **arc 2:** \( N/V \) from *fido* to *ate*  
  \[ \text{ate} \]  
  [5.1.15]

- **arc 3:** \( V\backslash N \) from *eat* to *bone*  
  \[ \text{the\_bone} \]  
  [5.1.16]

- **arc 4:** \( N\backslash N \) from *bone* to *table*  
  \[ \text{on\_the\_table} \]  
  [5.1.17]

- **arc 5:** \( N\backslash N \) from *table* to *tree*  
  \[ \text{under\_the\_tree} \]  
  [5.1.18]

- **arc 6:** \( N\backslash N \) from *tree* to *garden*  
  \[ \text{in\_the\_garden} \]  
  [5.1.19]

- **arc 7:** \( N\uparrow N \) from *garden* to *tree*  
  [5.1.20]

- **arc 8:** \( N\uparrow N \) from *tree* to *table*  
  [5.1.21]

- **arc 9:** \( N\uparrow N \) from *table* to *bone*  
  [5.1.22]

- **arc 10:** \( N\nwarrow V \) from *bone* to *eat*  
  [5.1.23]

The next item on the to-do list [1.5.1] is point 6, i.e. the sequence of explicit speak mode operations. The first four applications resemble [3.2.13] [3.2.17].

5.1.14 **NAVIGATING WITH V/N FROM eat to fido (arc 1)**

\[
\begin{align*}
\text{pattern level} & \\
V/N(1) & \\
\text{verb: } & \alpha \\
\text{arg: } & \beta \ X \\
\text{prn: } & K \\
\hline
\text{content level} & \\
\text{sur: } & \text{lexnoun}(\hat{\beta}) \\
\text{verb: } & \alpha \\
\text{cat: } & \#n' \ #a' \ \text{decl} \\
\text{sem: } & \text{past} \\
\text{arg: } & (\text{dog } x ) \ \text{bone} \\
\text{mdr: } & \\
\text{nc: } & \\
\text{pc: } & \\
\text{prn: } & 25
\end{align*}
\]

\[ \Rightarrow \begin{align*}
\text{sur: } & \text{noun}(\beta) \\
\text{noun: } & \beta \\
\text{fnc: } & \alpha \\
\text{prn: } & K
\end{align*} \]

#-mark \( \beta \) in the arg slot of proplet \( \alpha \)

5.1.15 **NAVIGATING WITH N/V FROM fido BACK TO eat (arc 2)**

\[
\begin{align*}
\text{pattern level} & \\
N/V(2) & \\
\text{noun: } & \beta \\
\text{fnc: } & \alpha \\
\text{mdr: } & Z \\
\text{prn: } & K \\
\hline
\text{content level} & \\
\text{sur: } & \text{fido} \\
\text{noun: } & (\text{dog } x) \\
\text{cat: } & \text{snp} \\
\text{sem: } & \text{nm} \ f \\
\text{fnc: } & \text{eat} \\
\text{mdr: } & \\
\text{nc: } & \\
\text{pc: } & \\
\text{prn: } & 25
\end{align*}
\]

\[ \Rightarrow \begin{align*}
\text{sur: } & \text{lexverb}(\hat{\alpha}) \\
\text{verb: } & \alpha \\
\text{arg: } & \#\beta \ Y \\
\text{prn: } & K
\end{align*} \]

#-mark \( \alpha \) in the fnc slot of proplet \( \beta \)

\( Z \) is NIL, or elementary and #-marked.
5.1.16 Navigating with $\mathcal{V}\setminus\mathcal{N}$ from \textit{eat} to \textit{bone} (arc 3)

\begin{equation}
\mathcal{V}\setminus\mathcal{N} (\mathcal{I})
\end{equation}

\begin{align*}
\text{pattern level} & : \\
\text{verb: } & \alpha \\
\text{arg: } & \#X \beta \ Y \\
\text{prn: } & K \\
\Rightarrow & \\
\text{content level} & : \\
\text{sur: } & \text{lexnoun(}\hat{\beta}\text{)} \\
\text{noun: } & \beta \\
\text{fnc: } & \alpha \\
\text{prn: } & K \quad \text{#-mark } \beta \text{ in the arg slot of proplet } \alpha
\end{align*}

The following three navigation steps move from \textit{bone} to \textit{table}, from \textit{table} to \textit{tree}, and from \textit{tree} to \textit{garden}:

5.1.17 Navigating with $\mathcal{N}\setminus\mathcal{N}$ from \textit{bone} to \textit{table} (arc 4)

\begin{equation}
\mathcal{N}\setminus\mathcal{N} (\mathcal{I})
\end{equation}

\begin{align*}
\text{pattern level} & : \\
\text{noun: } & \alpha \\
\text{mdr: } & \beta \\
\text{prn: } & K \\
\Rightarrow & \\
\text{content level} & : \\
\text{sur: } & \text{on the table} \\
\text{noun: } & \text{table} \\
\text{cat: } & \text{adnv snp} \\
\text{sem: } & \text{on def sg} \\
\text{fnc: } & \#\text{eat} \\
\text{mdr: } & \text{table} \\
\text{nc: } & \\
\text{pc: } & 25 \\
\text{prn: } & K \quad \text{#-mark } \beta \text{ in the mdr slot of proplet } \alpha
\end{align*}

5.1.18 Navigating with $\mathcal{N}\setminus\mathcal{N}$ from \textit{table} to \textit{tree} (arc 5)

\begin{equation}
\mathcal{N}\setminus\mathcal{N} (\mathcal{I})
\end{equation}

\begin{align*}
\text{pattern level} & : \\
\text{noun: } & \alpha \\
\text{mdr: } & \beta \\
\text{prn: } & K \\
\Rightarrow & \\
\text{content level} & : \\
\text{sur: } & \text{under the tree} \\
\text{noun: } & \text{tree} \\
\text{cat: } & \text{adnv snp} \\
\text{sem: } & \text{under def sg} \\
\text{fnc: } & \#\text{bone} \\
\text{mdr: } & \text{garden} \\
\text{nc: } & \\
\text{pc: } & 25 \\
\text{prn: } & K \quad \text{#-mark } \beta \text{ in the mdr slot of proplet } \alpha
\end{align*}
5.1.19 Navigating with N↓N from tree to garden (arc 6)

**N↓N**

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: α</td>
<td>sur: lexnoun(β)</td>
</tr>
<tr>
<td>mdr: β</td>
<td>noun: β</td>
</tr>
<tr>
<td>prn: K</td>
<td>mdd: α</td>
</tr>
<tr>
<td></td>
<td>prn: K</td>
</tr>
<tr>
<td></td>
<td>#-mark in the mdr slot of proplet α</td>
</tr>
</tbody>
</table>

The following navigation steps from garden back to tree (arc 7), from tree back to table (arc 8), and from table back to bone (arc 9) are empty because of #-markings assigned during the traversals of arcs 3–6.

5.1.20 Navigating with N↑N from garden back to tree (arc 7)

**N↑N**

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: α</td>
<td>sur: lexnoun(β)</td>
</tr>
<tr>
<td>mdd: β</td>
<td>noun: β</td>
</tr>
<tr>
<td>prn: K</td>
<td>mdr: α</td>
</tr>
<tr>
<td></td>
<td>prn: K</td>
</tr>
<tr>
<td></td>
<td>#-mark in the sur slot of proplet α</td>
</tr>
</tbody>
</table>

5.1.21 Navigating with N↑N from tree back to table (arc 8)

**N↑N**

<table>
<thead>
<tr>
<th>pattern level</th>
<th>content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: α</td>
<td>sur: lexnoun(β)</td>
</tr>
<tr>
<td>mdd: β</td>
<td>noun: β</td>
</tr>
<tr>
<td>prn: K</td>
<td>mdr: α</td>
</tr>
<tr>
<td></td>
<td>prn: K</td>
</tr>
<tr>
<td></td>
<td>#-mark in the sur slot of proplet α</td>
</tr>
</tbody>
</table>
5.1.22 Navigating with N↑N from table back to bone (arc 9)

The speak mode operations N↓N and N↑N traversing the initial modification of bone (5.1.17, 5.1.22) and those traversing the repetitions (5.1.18–5.1.21) are the same, and correspondingly for N×PREP in the hear mode.

The final traversal is from bone back to the top verb eat:

5.1.23 Navigating with N\V from bone back to eat (arc 10)

Using the cat value decl of the goal proplet, lexverb realizes the surface.

Following standard procedure, we conclude with point 7 on the to-do list. Extending DBS.18 to an example class of repeating prepositional phrases required the definition of the hear operation N×PREP (h24), and of the two speak mode operations N↓N (s19) and N↑N (s20).

Including the three lines which were omitted at the beginning of 5.1.2 for reasons of space, the hear mode derivation used 13 operation applications, while the speak mode derivation used ten:
5.1.24 Comparing Hear and Speak Mode Surface Segmentation

**Hear mode:**

1. Fido
2. ate
3. the
4. bone
5. on
6. the
7. table
8. under
9. the
10. tree
11. in
12. the
13. garden.

**Speak mode:**

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

Even though the speak mode requires two navigation steps for each modifier, i.e. 4 and 7, 5 and 8, and 6 and 9 (5.1.12), it has fewer applications than the hear mode because the three words of the prepositional phrases are each produced in one step from a single proplet by the lexicalization rule. The hear mode, in contrast, needs three applications for attaching and composing each prepositional phrase, i.e. 4-5-6, 7-8-9, and 10-11-12.

5.2 DBS.20: Repeating Subject Gapping

Gapping is a repeating construction in which the re-occurrences of a *shared item* are treated as gaps in the surface, while the semantic relations in the content are complete, as shown by the following content of a subject gapping:

5.2.1 Bob bought an apple, ∅ peeled a pear, and ∅ ate a peach.
In the content, the proplet *bob* serves as the shared subject of the predicates *buy*, *peel*, *eat* by specifying them in its `fnc` slot as the *gap list*. The inverse relation from the predicates to their shared subject is established by writing the core value *(person x)* of *bob* into the first *(subject)* `arg` slot of the verbs.

These relations are coded into lexical proplets by the following strictly time-linear, surface compositional hear mode derivation:\(^3\)

### 5.2.2 Graphical Hear Mode Derivation of the Content 5.2.1

<table>
<thead>
<tr>
<th>Step</th>
<th>Proplet</th>
<th>Surface</th>
<th>Syntactic-Semantic Parsing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>bob</em></td>
<td>Bob bought an apple peeled a pear and ate a peach.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><em>bob</em></td>
<td>cross-copying</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><em>bob</em></td>
<td>absorption with simultaneous substitution</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><em>bob</em></td>
<td>cross-copying</td>
<td></td>
</tr>
</tbody>
</table>

\(^3\) Gapping and suspension are complementary as follows: In the hear mode derivation of a gapping construction, the same proplet at the now front is used by the same operation in different derivation steps as first input. In a suspension compensation such as 2.5.2 lines 2a and 2b, or 2.6.2 lines 4a and 4b, different proplets are used as first input by different operations in the same derivation step.
The derivation of the content [5.2.1] in [5.2.2] is based on the method of how DBS hear mode operations apply in general (NLC Sect. 11.3): the current next word proplet activates all operations which match it with their second input pattern; the activated operations look at the now front for proplets matching their first input pattern and apply.
Thereby, a proplet at the current now front may be concatenated more than once. For example, the hear mode derivation of a transitive declarative sentence (2.3.2, 2.4.2) first uses the predicate for concatenation with the subject and then for concatenation with the grammatical object.

Because gapping is an intrapropositional construction (shown by the single prn value 32 in 5.2.1 and 5.2.2), the shared subject proplet *bob* is still at the now front after it has been combined with *bought* by SBJ×PRD in line 1. When the other compatible verbs *peel* and *eat* arrive, *bob* is re-used as their shared subject (lines 4 and 8b).

Subject gapping is special insofar as (i) the core values of the additional verbs are stored in the fnč slot of shared subject *bob* as the gap list and (ii) the core value of *bob* is copied into the initial (subject) arg slot of the additional verbs. Because the few proplets at the current now front are content-addressable, i.e. accessed by pattern and not by location (CLaTR Sect. 4.1), the apparent distance between proplets serving as input to cross-copying, e.g. line 8b, presents no obstacle.

The counterpart to multiple concatenations in the hear mode are multiple traversals in the speak mode. For example, the selective activation of a transitive declarative sentence first traverses the predicate to retrieve the continuation value for navigating to the subject and then returns to the predicate to retrieve the continuation value for navigating to the object (e.g. 2.3.6, 2.4.10). Similarly in subject gapping (5.2.15), which uses the shared subject repeatedly for finding the next item on the gap list as the continuation value.

In accordance with the Continuity Condition (NLC 3.6.5), the DBS.Think-Speak navigation along the semantic relations between proplets is continuous even in gapping constructions. This is possible by leaving the control of the gaps in the surface to the lexicalization rules. For example, in the case of subject gapping, lexnoun realizes the surface of the shared noun if, and only if, its initial fnč value is not yet #-marked (compare 5.2.17 with 5.2.21 and 5.2.25).

That not every traversal of a proplet results in the realization of a surface is by no means limited to gapping, but may occur in all constructions of natural language. This is shown by point 5 of the DBS laboratory set up 1.5.1, exemplified by 3.1.12, 3.2.12, 3.3.10, 3.4.11, 3.5.11, and many more.

A content is relatively language independent because DBS uses language-dependent surfaces solely as the input to the automatic word form recognition of the hear mode and as the output of automatic word form production in

---

4 For example, apart from the surfaces, the contents representing *The little dog found a big bone* in English, German, French, Italian, Spanish, Polish, Russian, etc. are pretty much the same sets of proplets in DBS (NLC Sect 4.6, CLaTR Sect. 3.6).
the speak mode. Processing content independently of any language surfaces is important for a unified control of the agent’s behavior, with and without language, by means of inferencing.

We follow standard procedure and continue with point 3 on the to-do list (e.g. 2.1.2), i.e. the complete sequence of explicit hear mode operation applications. The beginning of example 5.2.1 is a standard declarative sentence with a two-place verb. Therefore, the operations applying in lines 1–3 are SBJ x PRD (e.g. 2.1.2), PRD x OBJ (e.g. 2.3.4), and DET x CN (e.g. 2.4.8):

5.2.3 CROSS-COPYING Bob AND bought WITH SBJ x PRD (line 1)

5.2.4 CROSS-COPYING buy AND a WITH PRD x OBJ (line 2)

---

5 Outside the DBS laboratory set-up (2.5.1), successful behavior control is evaluated in terms of the agent’s survival vis à vis a constantly changing external and internal environment (CLaTR Chap. 6).
5.2.5 **A**bsorbing apple INTO _n_1 WITH \(\text{DET} \cup \text{CN}\) (line 3)

\[
\begin{align*}
\text{pattern} & \quad \text{content} \\
\text{level} & \quad \text{level}
\end{align*}
\]

At this point, the time-linear hear mode derivation has interpreted a regular declarative sentence with a transitive verb, i.e. Bob bought an apple.

The next proplet provided by automatic word form recognition is peel. \(\text{SBJ} \times \text{PRD}\) can not apply because the \(\text{fnc}\) slot of the shared subject bob is already occupied by the value buy. Therefore, we define the new operation \(\text{SBJ}^S \times \text{PRD}\), with the superscript \(S\) showing the subject to be the shared item:

5.2.6 **C**ross-copying bob AND peeled WITH \(\text{SBJ}^S \times \text{PRD}\) (line 4)

\[
\begin{align*}
\text{pattern} & \quad \text{content} \\
\text{level} & \quad \text{level}
\end{align*}
\]

\(\text{SBJ}^S \times \text{PRD}\) cross-copies the core value of the shared subject into the first (subject) \(\text{arg}\) slot of the next, here second, predicate and adds the core value of the second predicate to the \(\text{fnc}\) slot of the subject. The variable \(X\) in the feature [fnc: \(X\)] is used to buffer the presence of one or more values, here buy, already in place.
Every time SBJ$^\text{a}$×PRD applies, the gap list of the shared subject's fnc slot is extended by another verb. The formatting of the gap list in the noun proplet of the output, here writing the fnc value of peel underneath rather than beside the fnc value of buy, is left to the procedural side of the software.

With the first application of SBJ$^\text{a}$×PRD, the time-linear hear mode derivation has interpreted Bob bought an apple, $\emptyset$ peeled. The next word provided by automatic word form recognition is the determiner a(n), which is connected to peel with standard PRD×OBJ:

5.2.7 CROSS-COPYING peel AND a(n) WITH PRD×OBJ (line 5)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
<th>Pattern Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb: $\beta$</td>
<td>Sur: peel</td>
<td>Verb: $\beta$</td>
</tr>
<tr>
<td>Cat: #X N' Y $\gamma$</td>
<td>Noun: a(n)</td>
<td>Cat: #X N' Y $\gamma$</td>
</tr>
<tr>
<td>Arg: Z</td>
<td>Noun: n_2</td>
<td>Arg: (person x)</td>
</tr>
<tr>
<td>Prn: K</td>
<td>Prn: 32</td>
<td>Prn: K</td>
</tr>
</tbody>
</table>

The first gapped item is completed by adding pear with standard DET∪CN:

5.2.8 ABSORBING pear INTO n_2 WITH DET∪CN (line 6)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
<th>Pattern Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun: N_n</td>
<td>Sur: pear</td>
<td>Noun: \alpha</td>
</tr>
<tr>
<td>Cat: X CN' NP</td>
<td>Cat: sn</td>
<td>Cat: X NP</td>
</tr>
<tr>
<td>Sem: Y</td>
<td>Cat: sn'</td>
<td>Sem: Y Z</td>
</tr>
<tr>
<td>Prn: K</td>
<td>Prn: 32</td>
<td>Prn: K</td>
</tr>
</tbody>
</table>

The hear mode derivation has now interpreted Bob bought an apple, $\emptyset$ peeled a pear.
With the completion of the first gapped item, the second may begin. Theoretically, adding gapped items may go on indefinitely. Here, however, automatic word form recognition provides a coordinating conjunction, which signals to the hearer the imminent arrival of the final gapped item.

Because there is no semantic relation between *pear* and the coordinating conjunction *and*, the conjunction is suspended until its conjunct *ate* arrives. The suspension operation is $N\sim CNJ$ (3.6.4):

5.2.9 **Suspending** *pear* **and** with $N\sim CNJ$ (line 7)

\[
\begin{array}{l}
\text{pattern} \\
\text{level} \\
\begin{array}{c}
\text{noun: } \alpha \\
\text{sem: } X \\
\text{prn: } K
\end{array} \quad \begin{array}{c}
\text{verb: } V_n \\
\text{sem: } \beta \\
\text{prn: }
\end{array} \quad \begin{array}{c}
\text{noun: } \alpha \\
\text{sem: } X \\
\text{prn: } K
\end{array} \\
\beta \in \{\text{and, or}\}
\end{array}
\]

\[
\begin{array}{l}
\text{content} \\
\text{level} \\
\begin{array}{c}
\text{sur: pear} \\
\text{cat: snp} \\
\text{sem: indef sg} \\
\text{fnc: peel} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: 32}
\end{array} \quad \begin{array}{c}
\text{sur: } \text{and} \\
\text{cat: } \text{and} \\
\text{sem: } \text{and} \\
\text{fnc: peel} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: 32}
\end{array} \\
\begin{array}{c}
\text{sur: } \text{ate} \\
\text{cat: } \text{eat} \\
\text{sem: past} \\
\text{arg:} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn:}
\end{array} \\
\begin{array}{c}
\text{sur: } \text{and} \\
\text{cat: } \text{and} \\
\text{sem: } \text{and} \\
\text{fnc: peel} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: 32}
\end{array} \\
\begin{array}{c}
\text{sur: } \text{ate} \\
\text{cat: } \text{eat} \\
\text{sem: past} \\
\text{arg:} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn:}
\end{array}
\end{array}
\]

Next, the *eat* proplet is provided by automatic word form recognition. It absorbs the conjunction and with $CNJ \cup V$:

5.2.10 **Absorbing** *and* **into** *eat* **with** $CNJ \cup V$ (line 8a)

\[
\begin{array}{l}
\text{pattern} \\
\text{level} \\
\begin{array}{c}
\text{verb: } V_n \\
\text{sem: } \beta \\
\text{prn: } K
\end{array} \quad \begin{array}{c}
\text{verb: } \alpha \\
\text{sem: } X \\
\text{prn: }
\end{array} \\
\beta \in \{\text{and, or}\}
\end{array}
\]

\[
\begin{array}{l}
\text{content} \\
\text{level} \\
\begin{array}{c}
\text{sur: } \text{v}_1 \\
\text{cat: } \text{n}'a'v \\
\text{sem: } \text{and} \\
\text{fnc: } \text{past} \\
\text{arg:} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: 32}
\end{array} \\
\begin{array}{c}
\text{sur: } \text{ate} \\
\text{cat: } \text{eat} \\
\text{sem: past} \\
\text{arg:} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn:}
\end{array}
\end{array}
\]

To allow a general application of the operation $SBJ^s \times PRD$, regardless of whether or not the current output contains a conjunction, $CNJ \cup V$ does not provide a prn value to the output proplet, similar to $CNJ \cup N$ (4.39).
Once the conjunction has been absorbed, the input of the operation is replaced by its output and \( SBJ^* \times PRD \) (5.2.6) applies for the second and last time.

5.2.11 **Cross-copying bob and ate with \( SBJ^* \times PRD \) (line 8b)**

![Pattern diagram]

The gap list in the shared subject **bob** is now **buy peel eat** and the new predicate **eat** has the initial (subject) arg value (person x) of the shared subject. Because \( SBJ^* \times PRD \) cancels the subject valency position \( n' \) in the cat slot of **eat** by \(^{-}\#\)-marking, standard **PRD \times OBJ** may add (the beginning of) the grammatical object, provided by automatic word form recognition:

5.2.12 **Cross-copying eat and a(n) with \( PRD \times OBJ \) (line 9)**

![Pattern diagram]

It remains to complete the object with \( DET \cup CN \):
5.2.13 **ABSORBING peach INTO \( n_4 \) WITH \( \text{DET} \cup \text{CN} \) (line 10)**

The final step of the hear mode derivation is adding the period with \( \text{S} \cup \text{IP} \). However, the current now front happens to contain three verb proplets, namely *buy*, *peel*, and *eat*. Which of them should \( \text{S} \cup \text{IP} \) take as its first input proplet?

In subject and object gapping (Sect. 5.4) neither of the several verbs can claim to be the unique top predicate. The first verb, here *buy*, is most natural to enter the construction and to represent the proposition in the extrapositional coordination (1.4.4). The final verb, here *eat*, is most natural to exit the construction because it provides the shortest route to the verb of the next proposition (1.4.5–1.4.7).

The insertion of the conjunction, here *and* (5.2.2 line 8a), makes the last verb in a subject gapping construction identifiable as the last conjunct. When the second input pattern of \( \text{S} \cup \text{IP} \) is activated and there is a choice of verbs at the now front, it selects the one with the conjunction:

5.2.14 **ABSORBING • INTO *eat* WITH \( \text{S} \cup \text{IP} \) (LINE 11)**
Following standard procedure, let us continue with point 4 of 1.5.1.

5.2.15 **Graph analysis underlying production of 5.2.1**

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

The different tilts of the three N/V lines are solely for visual separation in the graph. The gaps appear as the empty traversals of the arcs 4, 1, and 8, 5. The navigation ends with arc 11. The upward arc 9 does not have a downward counterpart. The arc numbering is breadth first (Sect. 1.4).

We continue with point 5 on the to-do list 1.5.1 followed by point 6:

5.2.16 **Sequence of operation names and surface realizations**

<table>
<thead>
<tr>
<th>Arc</th>
<th>Operation</th>
<th>Name</th>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V/N from buy to bob</td>
<td>Bob</td>
<td>5.2.17</td>
</tr>
<tr>
<td>2</td>
<td>N/V from bob to buy</td>
<td>bought</td>
<td>5.2.18</td>
</tr>
<tr>
<td>3</td>
<td>V/N from buy to apple</td>
<td>an_apple</td>
<td>5.2.19</td>
</tr>
<tr>
<td>4</td>
<td>V/N from apple to buy</td>
<td></td>
<td>5.2.20</td>
</tr>
<tr>
<td>5</td>
<td>V/N from buy to bob</td>
<td></td>
<td>5.2.21</td>
</tr>
<tr>
<td>6</td>
<td>N/V from bob to peel</td>
<td>peeled</td>
<td>5.2.22</td>
</tr>
<tr>
<td>7</td>
<td>V/N from peel to pear</td>
<td>a_pear</td>
<td>5.2.23</td>
</tr>
<tr>
<td>8</td>
<td>N/V from pear to peel</td>
<td></td>
<td>5.2.24</td>
</tr>
<tr>
<td>9</td>
<td>V/N from peel to bob</td>
<td></td>
<td>5.2.25</td>
</tr>
<tr>
<td>10</td>
<td>N/V from bob to eat</td>
<td>and_ate</td>
<td>5.2.26</td>
</tr>
<tr>
<td>11</td>
<td>N/V from eat to peach</td>
<td>a_peach</td>
<td>5.2.27</td>
</tr>
<tr>
<td></td>
<td>V/N from peach to eat</td>
<td></td>
<td>5.2.28</td>
</tr>
</tbody>
</table>
The first speak mode operation to apply is new V/N\(^i\). The superscript \(^i\) characterizes the initial subject as the shared item. It differs from standard V/N because it accepts a gap list in the fnc slot of the goal proplet. V/N\(^i\) differs from V/N\(^s\) \((5.2.21, 5.2.25)\) because it has an instruction to #-mark the shared subject, here the arg value (person x), which is absent in V/N\(^s\).

5.2.17 NAVIGATING WITH V/N\(^i\) FROM buy TO bob (arc 1)

\[
\begin{align*}
\text{pattern level} & \quad \left[
\begin{array}{c}
\text{verb: } \alpha \\
\text{arg: } \beta X \\
\text{prn: } K
\end{array}
\right] \quad \Rightarrow \quad \left[
\begin{array}{c}
\text{sur: lexnoun(} \hat{\beta} \text{)} \\
\text{nou}\_\text{m: } \beta \\
\text{fnc: } \alpha Z \\
\text{prn: } K
\end{array}
\right]
\end{align*}
\]

\#-mark \(\beta\) in the arg slot of proplet \(\alpha\)

The result of the instruction shows up in the goal proplet of 5.2.18.

For the return to the verb proplet corresponding to \(\alpha\) in 5.2.17, we define the new operation N\(^s\)/V. It differs from standard N\(^s\)/V in that it accepts a subject noun with a gap list. The instruction of N\(^s\)/V incrementally #-marks the currently first unmarked fnc value in the gap list, one at each application.

5.2.18 NAVIGATING WITH N\(^s\)/V FROM bob BACK TO buy (arc 2)

\[
\begin{align*}
\text{pattern level} & \quad \left[
\begin{array}{c}
\text{nou}\_\text{m: } \beta \\
\text{fnc: #X } \alpha Y \\
\text{mdr: } Z \\
\text{prn: } K
\end{array}
\right] \quad \Rightarrow \quad \left[
\begin{array}{c}
\text{sur: lexverb(} \hat{\alpha} \text{)} \\
\text{verb: } \alpha \\
\text{arg: } \beta W \\
\text{prn: } K
\end{array}
\right]
\end{align*}
\]

\#-mark \(\alpha\) in the fnc slot of proplet \(\beta\)

\(Z\) is #-marked or NIL.

\[
\begin{align*}
\text{content level} & \quad \left[
\begin{array}{c}
\text{sur: bob} \\
\text{nou}\_\text{m: (person x)} \\
\text{cat: snp} \\
\text{sem: nm m} \\
\text{fnc: buy} \\
\text{peel} \\
\text{eat} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: } 32
\end{array}
\right] \quad \Downarrow \quad \left[
\begin{array}{c}
\text{sur: bought} \\
\text{verb: buy} \\
\text{cat: #n' } \#a' \text{ decl ip+} \\
\text{sem: past} \\
\text{arg: #(person x) apple} \\
\text{mdr:} \\
\text{nc:} \\
\text{pc:} \\
\text{prn: } 32
\end{array}
\right]
\end{align*}
\]
The first unmarked item on the gap list, here *buy*, provides the address of the goal proplet. Because *buy* is initial, the variable #X is assigned the value NIL. Lexverb realizes the predicate surface *bought* based on the core value *buy*, the initial cat value *n*, and the sem value *past*. The result of the instruction will show up in the output of 5.2.21.

The second arg value *apple* of *buy* provides the goal proplet for \( V \rightarrow_N \)

**5.2.19 NAVIGATING WITH \( V \rightarrow_N \) FROM *buy* TO *apple* (arc 3)**

\[
\begin{align*}
\text{pattern level} & \quad \text{content level} \\
\begin{array}{c}
\text{verb: } \alpha \\
\text{arg: } #X \beta Y \\
\text{prn: } K
\end{array} & \quad \begin{array}{c}
\text{sur: lexnoun(}\beta\text{)} \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K
\end{array} \quad \Rightarrow \quad \begin{array}{c}
\text{sur: an_apple} \\
\text{noun: apple} \\
\text{cat: snp} \\
\text{sem: indef sg} \\
\text{fnc: } \#\text{buy} \\
\text{mdr: } Z \\
\text{nc: } \text{pc: } \text{prn: 32}
\end{array} \quad \begin{array}{c}
\text{sur: lexverb(}\alpha\text{)} \\
\text{verb: } \alpha \\
\text{arg: } #X \beta Y \\
\text{prn: } K
\end{array}
\end{align*}
\]

The initial sentence is completed by returning to the verb with standard \( N \rightarrow_V \).

**5.2.20 NAVIGATING WITH \( N \rightarrow_V \) FROM *apple* TO *buy* (arc 4)**

\[
\begin{align*}
\text{pattern level} & \quad \text{content level} \\
\begin{array}{c}
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K
\end{array} & \quad \begin{array}{c}
\text{sur: lexverb(}\alpha\text{)} \\
\text{verb: } \alpha \\
\text{arg: } #X \beta Y \\
\text{prn: } K
\end{array} \quad \Rightarrow \quad \begin{array}{c}
\text{sur: apple} \\
\text{cat: snp} \\
\text{sem: indef sg} \\
\text{fnc: } \#\text{buy} \\
\text{mdr: } \text{nc: } \text{pc: } \text{prn: 32}
\end{array} \quad \begin{array}{c}
\text{sur: } \text{buy} \\
\text{cat: } \#n \#a \text{ decl ip+} \\
\text{sem: } \text{past} \\
\text{arg: } #(\text{person x}) \text{apple} \\
\text{mdr: } Z \\
\text{nc: } \text{pc: } \text{prn: 32}
\end{array}
\end{align*}
\]

At this point, lexverb could have added a period. However, there are several untraversed proplets with the intrapropositional prn value 32 still waiting in the word bank to be traversed. To continue with a first gapped item, the navigation returns to the shared subject to find the first unmarked fnc value, here *peel*, in its gap list as the continuation value.
5.2.21 Navigating with \( V / N^S \) from \( buy \) to \( bob \) (arc 1)

The distinctive features of \( V / N^S \) are \[ \text{arg: } \#\beta \#X \] in the start and \[ \text{fnc: } \#Y \alpha Z \] in the goal proplet. The first feature ensures that the initial sentence has been completely traversed (5.2.17–5.2.20), the second that at least one item of the gap list has been canceled (5.2.18). Because the initial value on the gap list, i.e. \( buy \), is already #-marked, no subject surface is produced by lexnoun.

The speak mode derivation has now reached Bob bought an apple \( \emptyset \), with \( \emptyset \) indicating the first subject gap. The next item on the gap list, i.e. \( peel \), is accessed with the operation \( N^S / V \) (5.2.18).

5.2.22 Navigating with \( N^S / V \) from \( bob \) to \( peel \) (arc 6)

---

\( as \) in \( SBJ^S \times PRD \) (5.2.6), the \( S \) in \( N^S / V \) indicates the shared subject, though here in the speak mode.
Lexverb produces the surface peeled based on the #-marking of the first (subject) arg value, the core value peel, and the value past of the sem slot. The instruction #-marks peel on the gap list (showing up in 5.2.25).

Using the unmarked continuation value pear of peel, the standard V\N operation navigates to the object of the gapped item.

5.2.23 **Navigating with V\N from peel to pear (arc 7)**

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \\
\text{content level} & \quad \downarrow
\end{align*}
\]

Using the values pear and indef sg, lexnoun realizes a_pear.

Next, standard N\V returns to the predicate of the current gapped item:

5.2.24 **Navigating with N\V from pear to peel (arc 8)**

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \\
\text{content level} & \quad \downarrow
\end{align*}
\]

The transition is empty because the second arg value of peel is #-marked, indicating that the navigation is already in postverbal territory, and the final cat value is still v.

The speak mode derivation has now produced Bob bought an apple ∅ peeled a pear and has returned to peel. Using its arg value #(person x),
\(V^s/N^s\) moves from the verb of the current gapped item back to the shared subject:

5.2.25 **Navigating with** \(V^s/N^s\) **from** *peel* **to** *bob* (arc 5)

\[
\begin{align*}
\text{pattern level} & : \\
\verb: & \alpha \\
\arg: & \beta \#X \\
\text{pron:} & K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur:} & \text{lexnoun}(\beta) \\
\text{noun:} & \beta \\
\text{fnc:} & \#Y \alpha Z \\
\text{pron:} & K
\end{align*}
\]

\[
\begin{align*}
\text{content level} & : \\
\text{sur:} & \text{peel} \\
\text{cat:} & \#n' \#a' v \\
\text{sem:} & \text{past} \\
\text{arg:} & \text{#(person x) #pear} \\
\text{mdr:} & \\
\text{nc:} & \\
\text{pc:} & \text{prn: 32}
\end{align*}
\]

Next, \(N^s/V\) proceeds to the predicate of the new (and final) gapped item, here *eat*:

5.2.26 **Navigating with** \(N^s/V\) **from** *bob* **to** *eat* (arc 9)

\[
\begin{align*}
\text{pattern level} & : \\
\text{noun:} & \beta \\
\text{fnc:} & \#X \alpha Y \\
\text{mdr:} & Z \\
\text{pron:} & K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur:} & \text{lexverb}(\alpha) \\
\text{verb:} & \alpha \\
\text{arg:} & \beta W \\
\text{pron:} & K
\end{align*}
\]

\[
\begin{align*}
\text{content level} & : \\
\text{sur:} & \text{bob} \\
\text{noun:} & \text{(person x)} \\
\text{cat:} & \text{snp} \\
\text{sem:} & \text{nm m} \\
\text{fnc:} & \#buy \\
\text{#peel} \\
\text{#eat} \\
\text{mdr:} & \\
\text{nc:} & \\
\text{pc:} & \text{prn: 32}
\end{align*}
\]

Lexverb uses the #-marking of the first \texttt{arg} value (\texttt{person x}), the core value \texttt{eat}, and the \texttt{sem} values \texttt{and} \texttt{past} to realize the surface \texttt{and_ate}. The instruction #-marks the last item on the gap list.

Similar to 5.2.19 and 5.2.23, \(V \setminus N\) uses the unmarked continuation value in the \texttt{arg} slot of \texttt{eat} to navigate to \texttt{peach}:
5.2.27 Navigating with $V \backslash N$ from *eat* to *peach* (arc 10)

\[
\begin{align*}
\text{pattern level} & \quad \text{content level} \\
\text{verb: } \alpha & \quad \text{verb: } \alpha \\
\text{arg: } #X \beta Y & \quad \text{arg: } #X \beta Y \\
\text{prn: } K & \quad \text{prn: } K \\
[& \quad [\text{sur: lexnoun}(\beta)] \\
\uparrow & \quad \downarrow \\
[& \quad [\text{sur: peach}] \\
\text{cat: } #n\ ' #a' & \quad \text{cat: } #n\ ' #a' \\
\text{sem: } and \text{ past} & \quad \text{sem: } and \text{ past} \\
\text{arg: } #(\text{person } x) \text{ peach} & \quad \text{arg: } #(\text{person } x) \text{ peach} \\
\text{mdr: } & \quad \text{mdr: } \\
\text{nc: } & \quad \text{nc: } \\
\text{pc: } & \quad \text{pc: } \\
\text{prn: } 32 & \quad \text{prn: } 32 \\
\Rightarrow & \quad \Rightarrow \\
\end{align*}
\]

The operation traversing the relation between the predicate and the grammatical object of the gapped item is independent of the shared subject, like 5.2.23.

The speak mode derivation concludes with a return to the current verb to realize the period.

5.2.28 Navigating with $N \backslash V$ from *peach* to *eat* (arc 11)

\[
\begin{align*}
\text{pattern level} & \quad \text{content level} \\
\text{noun: } \beta & \quad \text{noun: } \beta \\
\text{fnc: } \alpha & \quad \text{fnc: } \alpha \\
\text{mdr: } Z & \quad \text{mdr: } Z \\
\text{prn: } K & \quad \text{prn: } K \\
[
\text{sur: lexverb}(\alpha)] & \quad [\text{sur: lexverb}(\alpha)] \\
\uparrow & \quad \downarrow \\
[\text{sur: peach}] & \quad [\text{sur: peach}] \\
\text{cat: } & \quad \text{cat: } \\
\text{sem: } & \quad \text{sem: } \\
\text{arg: } & \quad \text{arg: } \\
\text{mdr: } & \quad \text{mdr: } \\
\text{nc: } & \quad \text{nc: } \\
\text{pc: } & \quad \text{pc: } \\
\text{prn: } 32 & \quad \text{prn: } 32 \\
\Rightarrow & \quad \Rightarrow \\
\end{align*}
\]

In the speak mode, ending with arc 11 saves a lengthy return to the initial verb *buy* (5.2.15, 1.4.5) and similarly in 1.4.7. In the hear mode, the final verb *eat* is the natural place to place the nc value for continuing to a next proposition. It is also the easiest for $S \cup IP$ to find as its first input proplet.

Following standard procedure, we conclude with point 7 on the to-do list 1.5.1. Extending DBS.19 to an example class with subject gapping required the definition of the hear mode operation $SBJ \times PRD$ (14), and of the speak mode operations $V / N$ (37), $V / N$ (23) and $N / V$ (24). The hear and the speak mode derivation used 12 operation applications each.
5.2.29 Comparing Hear and Speak Mode Surface Segmentation

hear mode:

```
1  2  3  4  5  6  7
Bob × bought × an ∪ apple × peeled × a ∪ pear ×
```

and ∪ × ate × a ∪ peach ∪.

speak mode:

```
1  2  3  4  5  6
V/N Bob N^g/V bought V\N an_apple N\V V/N^g N^g/V peeled
```

The numbering of the speak mode operations corresponds to the arc traversal numbers in 5.2.14. Arc 1 is traversed twice.

5.3 DBS.21: Repeating Predicate Gapping

Just as subject gapping may be analyzed pretheoretically (NLC, Sect. 8.5) as

```
Bob buy apple
∅ peel pear
∅ eat peach
```

predicate gapping may be analyzed pretheoretically as

```
Bob buy apple
Jim ∅ pear
Bill ∅ peach
```

Consider the following surface of predicate gapping and its content:

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

The shared item is the predicate buy. Its arg slot contains the gap list, here the subject-object pairs bob apple, jim pear, and bill peach. The conjunction
and is coded into the initial sem slot of bill. The subject and object proplets take buy as their fnc value. Like all gapping constructions, predicate gapping is intrapropositional, as shown by the use of a single prn value, here 33.

Let us continue with point 2 on the to-do list

5.3.2 Graphical hear mode derivation of the content

5.3.1 syntactic−semantic parsing

<table>
<thead>
<tr>
<th>st</th>
<th>buy</th>
<th>noun: (p. x)</th>
<th>cat: sup</th>
<th>sem: nm m</th>
<th>fnc:</th>
<th>pm:</th>
<th>pm:</th>
<th>pm:</th>
<th>pm:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>buy</td>
<td>noun: (p. x)</td>
<td>cat: sup</td>
<td>sem: nm m</td>
<td>fnc:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
</tr>
<tr>
<td>2</td>
<td>buy</td>
<td>noun: (p. x)</td>
<td>cat: sup</td>
<td>sem: nm m</td>
<td>fnc:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
</tr>
<tr>
<td>3</td>
<td>buy</td>
<td>noun: (p. x)</td>
<td>cat: sup</td>
<td>sem: nm m</td>
<td>fnc:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
</tr>
<tr>
<td>4</td>
<td>buy</td>
<td>noun: (p. x)</td>
<td>cat: sup</td>
<td>sem: nm m</td>
<td>fnc:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
</tr>
<tr>
<td>5</td>
<td>buy</td>
<td>noun: (p. x)</td>
<td>cat: sup</td>
<td>sem: nm m</td>
<td>fnc:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
</tr>
<tr>
<td>6</td>
<td>buy</td>
<td>noun: (p. x)</td>
<td>cat: sup</td>
<td>sem: nm m</td>
<td>fnc:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
<td>pm:</td>
</tr>
</tbody>
</table>

absorption with simultaneous substitution
The surface of predicate gapping begins with a complete sentence, here the same as in the previous example 5.2.2 of subject gapping.

One direction of the relation between the shared predicate *buy* and the gapped items is coded by copying the core values of *jim pear* as the second line and the core values of *bill apple* as the third line into the arg slot of *buy* as the gap list. The other direction is coded by copying the core value of *buy* into the fnC slot of the subject and the object proplets of the first and the second gapped item (lines 4–6 and 8b–10).
In line 7, the conjunction *and*, provided by automatic word form recognition, is suspended. In line 8a, it is absorbed into the subject *bill* of the third gapped item. In line 11, the derivation concludes with the absorption of the punctuation mark into the shared item, which happens to be the sole predicate.

Let us follow standard procedure and continue with point 3 on the to-do list i.e. the complete sequence of explicit hear mode operation applications:

### 5.3.3 CROSS-COPYING Bob AND bought WITH SBJ×PRD (line 1)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agreement conditions as in 5.1.2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sur: Bob</th>
<th>sur: bought</th>
<th>sur: bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: (person x)</td>
<td>verb: buy</td>
<td>noun: (person x)</td>
</tr>
<tr>
<td>cat: snp</td>
<td>cat: n' a' v</td>
<td>cat: snp</td>
</tr>
<tr>
<td>sem: nm m</td>
<td>sem: past</td>
<td>sem: nm m</td>
</tr>
<tr>
<td>fnc: mdr</td>
<td>arg: fnc</td>
<td>mdr</td>
</tr>
<tr>
<td>nc: pc:</td>
<td>pc:</td>
<td>pc:</td>
</tr>
<tr>
<td>prn: 33</td>
<td>prn: 33</td>
<td>prn: 33</td>
</tr>
</tbody>
</table>

Computing possible continuations in the hear mode has the effect that a seemingly regular copying of the subject’s core value into the first arg slot of the predicate may turn out to be the beginning of the predicate’s gap list (5.3.1). The same holds for the following copying of the object’s core value into the predicate’s second arg slot:

### 5.3.4 CROSS-COPYING bought AND a(n) WITH PRD×OBJ (line 2)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: β cat: #X' N' Y γ</td>
<td>noun: α cat: CN' N</td>
</tr>
<tr>
<td>arg: Z</td>
<td>fnc:</td>
</tr>
<tr>
<td>prn: K</td>
<td>prn: K</td>
</tr>
<tr>
<td>Agreement conditions as in 5.4.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sur:</th>
<th>sur: a(n)</th>
<th>sur:</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: buy</td>
<td>noun: n_1</td>
<td>verb: buy</td>
</tr>
<tr>
<td>cat: n' a' v</td>
<td>fnc:</td>
<td>cat: n_1</td>
</tr>
<tr>
<td>sem: past</td>
<td>arg: (person x) n_1</td>
<td>sem: past</td>
</tr>
<tr>
<td>fnc:</td>
<td>mdr:</td>
<td>arg: (person x)</td>
</tr>
<tr>
<td>nc:</td>
<td>pc:</td>
<td>nc:</td>
</tr>
<tr>
<td>pc:</td>
<td>prn: 33</td>
<td>pc:</td>
</tr>
<tr>
<td>prn: 33</td>
<td>prn: 33</td>
<td>prn: 33</td>
</tr>
</tbody>
</table>

The grammatical object is completed by absorbing the common noun:
5.3.5 Absorbing apple into a(n) with $\text{DET} \cup \text{CN}$ (line 3)

\begin{align*}
\text{DET} \cup \text{CN} \quad \text{(5.3.7)}
\end{align*}

At this point, the time-linear hear mode derivation has interpreted the initial sentence Bob bought an apple.

Next, automatic word form recognition provides the first gapped item Jim $\emptyset$ a_pear, consisting of jim as the subject and pear as the grammatical object. The semantic interpretation of the gapped item requires the two nouns to be connected to the shared predicate $\text{buy}$ of the initial sentence by cross-copying.

For connecting the shared predicate to the subject of a gapped item, we define new $\text{PRD}^p \times \text{SBJ}$, with $^p$ showing the predicate to be shared item. It differs from standard $\text{SBJ} \times \text{PRD}$ in the order of the input items. Also, the #-marking of the object valency position in the cat attribute of the shared item $\text{buy}$ is removed in the output to enable the application of standard $\text{PRD} \times \text{OBJ}$ in 5.3.7 and 5.3.12.

5.3.6 Cross-copying buy and Jim with $\text{PRD}^p \times \text{SBJ}$ (line 4)

\begin{align*}
\text{PRD}^p \times \text{SBJ} \quad \text{(5.3.8)}
\end{align*}
The variable $Y$ in the $\text{arg}$ slot of the pattern matching the shared predicate ensures that $\text{arg}$ slot must not be empty (1.6.5).

For connecting the shared predicate and the grammatical object of the current gapped item, standard $\text{PRD} \times \text{OBJ}$ may be used:

5.3.7 CROSS-COPYING $buy$ AND $a(n)$ WITH $\text{PRD} \times \text{OBJ}$ (line 5)

The time-linear hear mode derivation has now interpreted Bob bought an apple, Jim $\emptyset$ a.

At this point, the object of the gapped item is represented by the substitution variable $n_2$ in the second position of the second gap list line of $buy$. It is replaced with the value pear by simultaneous substitution, induced by $\text{DET} \cup \text{CN}$ (cf. line 6 in 5.3.4):

5.3.8 ABSORBING pear INTO $a(n)$ WITH $\text{DET} \cup \text{CN}$ (line 6)
With the completion of the first gapped item, the second may begin. Theoretically, adding gapped items may go on indefinitely. Here, however, automatic word form recognition provides a coordinating conjunction, added with the suspension operation N~CNJ (5.2.9):

5.3.9 SUSPENDING pear and and with N~CNJ (line 7)

Next the suspended conjunction is absorbed into the first sem slot of the final gapped item’s subject:

5.3.10 ABSORBING and INTO Bill with CNJ∪N (line 8a)

The lexical entry of the conjunction uses the English surface, here and, to specify its kind. As the initial value of the sem attribute, it is (i) a significant contribution to the content and (ii) needed for realization in the speak mode (5.3.25). As in 3.6.5 and 5.2.10, no prn is assigned to the output.

The final gapped item is added with applications of the existing operations PRD×SBJ, PRD×OBJ, and DET∪CN:
5.3.11 Cross-copying *buy* and *bill* with PRD\(^\times\)OBJ (line 8b)

The operation cross-copies the core value (*person z*) of the gapped item’s subject into the third gap list line, and the core value of the shared predicate *buy* into the fnc slot of the third subject, i.e., the subject of the second gapped item. As in the previous application of PRD\(^\times\)SBJ in 5.3.6 the #-marking of X is removed in the output, enabling the addition of the current gapped item’s obligatory grammatical object with standard PRD\(^\times\)OBJ, which follows next:

5.3.12 Cross-copying *buy* and *a* with PRD\(^\times\)OBJ (line 9)

PRD\(^\times\)OBJ completes the third line of the gap list with the input feature [arg: Z] and the corresponding output feature [arg: Z \(\alpha\)]. Originally designed
for adding the object(s) of two- and three-place verbs (NLC Sect. 13.4), PRD \( \times \) OBJ serves here to extend the current gap list. The absorption of *peach* with \( \text{DET} \cup \text{CN} \) completes the grammatical object:

5.3.13 **Absorbing peach into \( n \_4 \) with \( \text{DET} \cup \text{CN} \) (line 10)**

\[
\begin{align*}
\text{DET} \cup \text{CN} & \quad \begin{cases}
\text{noun: } N_{n} \\
\text{cat: } X \text{ CN'} NP \\
\text{sem: } Y \\
\text{prn: } K
\end{cases} & \Rightarrow & \begin{cases}
\text{noun: } \alpha \\
\text{cat: } X \text{ NP} \\
\text{sem: } Y \ Z \\
\text{prn: } K
\end{cases} \\
\text{Agreement conditions as in 2.2.4}
\end{align*}
\]

The simultaneous substitution triggered by \( \text{DET} \cup \text{CN} \) replaces \( n \_4 \) in the second slot of the third gap list line in the shared predicate with the value *peach* (5.3.12).

For adding the period, the first input pattern of \( S \cup IP \) finds only a single predicate at the now front (in contradistinction to 5.2.14 and 5.4.12).

5.3.14 **Absorbing * into buy with \( S \cup IP \) (line 11)**

\[
\begin{align*}
S \cup IP & \quad \begin{cases}
\text{verb: } \beta \\
\text{cat: } \#X' \text{ VT} \\
\text{prn: } K
\end{cases} & \Rightarrow & \begin{cases}
\text{verb: } V_{n} \\
\text{cat: } VT' \text{ SM} \\
\text{prn: } K
\end{cases} & \Rightarrow & \begin{cases}
\text{verb: } \beta \\
\text{cat: } \#X' \text{ SM ip+} \\
\text{prn: } K
\end{cases} \\
\text{Agreement conditions as in 2.1.3}
\end{align*}
\]

The \text{cat} value \text{decl} of the period determines the sentential mood in the shared predicate proplet *buy*.
Similar to the linguistic analysis of subject gapping in Sect. 5.2, the time-linear hear mode derivation has shown the incremental build-up of the gap list in the shared item of a predicate gapping construction. For the complete set of proplets resulting from the hear mode derivation see 5.3.1.

Turning to point 4 on the to-do list 1.5.1, consider the DBS graph structure of the speak mode based on the content derived in 5.3.2:

### 5.3.15 Graph analysis underlying production of 5.3.1

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

![Graphs](image)

(iv) surface realization

```
Bob bought an_apple Jim a_pear and_Bill a_peach.
```

The shared predicate relates to the subject and object of its initial sentence (arcs 1–4) and of its two gapped items (arcs 5–8 and 9–12). As in subject gapping (5.2.14) and object gapping (5.4.13), the arc numbering is breadth-first: it follows the initial sentence and the gapped items from the outside in, completing each level before going to the next one down (Sect. 1.4).

In accordance with standard procedure, we continue with point 5 of 1.5.1, i.e. the list of speak mode operation names with surface realizations:

### 5.3.16 Sequence of operation names and surface realizations

<table>
<thead>
<tr>
<th>Arc</th>
<th>Operation</th>
<th>Surface Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>arc 1:</td>
<td>V\p/N from buy to bob</td>
<td>Bob</td>
</tr>
<tr>
<td>arc 2:</td>
<td>N\ V\ from bob to buy</td>
<td>bought</td>
</tr>
<tr>
<td>arc 3:</td>
<td>V\ N\ from buy to apple</td>
<td>an_apple</td>
</tr>
<tr>
<td>arc 4:</td>
<td>N\ V\ from apple to buy</td>
<td></td>
</tr>
<tr>
<td>arc 5:</td>
<td>V\p/N from buy to jim</td>
<td>Jim</td>
</tr>
<tr>
<td>arc 6:</td>
<td>N\ V\p from jim to buy</td>
<td></td>
</tr>
<tr>
<td>arc 7:</td>
<td>V\ N\ from buy to pear</td>
<td>a_pear</td>
</tr>
<tr>
<td>arc 8:</td>
<td>N\ V\ from pear to buy</td>
<td></td>
</tr>
<tr>
<td>arc 9:</td>
<td>V\p/N from buy to bill</td>
<td>and_Bill</td>
</tr>
<tr>
<td>arc 10:</td>
<td>N\ V\p from bill to buy</td>
<td></td>
</tr>
<tr>
<td>arc 11:</td>
<td>V\ N\ from buy to peach</td>
<td>a_peach</td>
</tr>
<tr>
<td>arc 12:</td>
<td>N\ V\ from peach to buy</td>
<td>.</td>
</tr>
</tbody>
</table>
The next item on the to-do list [1.5.1] is point 6, i.e. the sequence of explicit speak mode applications. The traversal of the initial sentence is the same as in [5.2.16], [5.2.19] (subject gapping), except that the gap list is now in the verb:

5.3.17 Navigating with \( \text{VP}_p/\text{N} \) from buy to bob (arc 1)

\[
\begin{align*}
\text{pattern level} & \\
\text{verb: } \alpha & \\
\text{arg: } \#X \beta \ Y & \\
\text{prn: } K & \\
\Rightarrow & \\
\text{content level} & \\
\text{sur: } \text{lexnoun}(\hat{\beta}) & \\
\text{verb: } \alpha & \\
\text{arg: } \#X \beta \ Y & \\
\text{prn: } K & \\
\text{#-mark } \beta & \text{ in the arg slot of proplet } \alpha
\end{align*}
\]

Compared to \( \text{V}/\text{N} \) (e.g. [2.1.6]), \( \text{VP}_p/\text{N} \) buffers the possibility of #-marked items preceding and unmarked items following \( \beta \) in the gap list with the feature [arg: \( \#X \beta \ Y \)] (here \( \#X \) is bound to \( \text{NIL} \)).

For the return to the verb proplet corresponding to \( \alpha \) in [5.3.17] we define the new operation \( \text{N}^{\dagger}/\text{V}^i \).

5.3.18 Navigating with \( \text{N}^{\dagger}/\text{V}^i \) from bob back to buy (arc 2)

\[
\begin{align*}
\text{pattern level} & \\
\text{noun: } \beta & \\
\text{fnc: } \alpha & \\
\text{mdr: } Z & \\
\text{prn: } K & \\
\Rightarrow & \\
\text{content level} & \\
\text{sur: } \text{bought} & \\
\text{verb: } \text{buy} & \\
\text{cat: } \#n \ #a \# \text{decl +ip} & \\
\text{sem: } \text{past} & \\
\text{arg: } \#(\text{person } x) \text{ apple} & \\
\text{name: } \text{person } y \text{ pear} & \\
\text{name: } \text{person } z \text{ peach} & \\
\text{prn: } 33 & \\
\end{align*}
\]

\( \text{N}^{\dagger}/\text{V}^i \) differs from standard \( \text{N}^{\dagger}/\text{V} \) in that it accepts a predicate with a gap list. Once the shared predicate has been realized, traversals from a subject to the
shared predicate use $N \backslash V^p$ (26); it differs from $N \backslash V^j$ because it has no instruction. First, however, the object part of the gapped item has to be realized:

5.3.19 **Navigating with $V \backslash N$ from buy to apple** (arc 3)

The variables $\#X$ and $Y$ in the feature $[\text{arg}: \#X \beta Y]$, originally introduced for accommodating oblique arguments (e.g. 2.3.10), serve here to accommodate the objects in the gap list cost-free. The initial sentence is completed by returning to the verb with $N \backslash V$.

5.3.20 **Navigating with $N \backslash V$ from apple to buy** (arc 4)

The navigation has now traversed the initial sentence Bob bought an apple and #-marked the two arguments in the shared predicate’s first gap list line.
The next step is to navigate from the shared predicate $\text{buy}$ to the first unmarked item in the gap list, i.e. to the subject $\text{jim}$ in the second line, using $V^p/N$ 

\[5.3.21\]

**Navigating with $V^p/N$ from $\text{buy}$ to $\text{jim}$ (arc 5)**

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{sur: lexnoun}(\hat{\beta})$</td>
<td>$\text{sur: jim}$</td>
</tr>
<tr>
<td>$\text{noun: } \beta$</td>
<td>$\text{noun: (person y)}$</td>
</tr>
<tr>
<td>$\text{fn}: \alpha$</td>
<td>$\text{cat: } \text{snp}$</td>
</tr>
<tr>
<td>$\text{prn: } K$</td>
<td>$\text{sem: } \text{nm } \text{m}$</td>
</tr>
<tr>
<td>$\text{arg: } #X \beta \ Y$</td>
<td>$\text{arg: } (\text{person } x) \ #\text{apple}$</td>
</tr>
<tr>
<td>$\text{sem: } \text{past}$</td>
<td>$\text{sem: } \text{nm } \text{m}$</td>
</tr>
<tr>
<td>$\text{mdr:}$</td>
<td>$\text{fnc: } #\text{buy}$</td>
</tr>
<tr>
<td>$\text{nc:}$</td>
<td>$\text{mdr:}$</td>
</tr>
<tr>
<td>$\text{pc:}$</td>
<td>$\text{nc:}$</td>
</tr>
<tr>
<td>$\text{prn: } 33$</td>
<td>$\text{pc:}$</td>
</tr>
</tbody>
</table>

Z is #-marked or NIL.

To find the next unmarked item, i.e. the gapped item’s object, in the predicate’s current gap list, the navigation must return from the gapped item’s subject to the shared predicate, using $N/V^p$:

\[5.3.22\]

**Navigating with $N/V^p$ from $\text{jim}$ back to $\text{buy}$ (arc 6)**

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{sur: lexverb}(\hat{\alpha})$</td>
<td>$\text{sur: buy}$</td>
</tr>
<tr>
<td>$\text{verb: } \alpha$</td>
<td>$\text{verb: } \text{buy}$</td>
</tr>
<tr>
<td>$\text{arg: } #X \ #\alpha \ #\beta \ Y$</td>
<td>$\text{cat: } #\text{a’} \ #\text{decl } +\text{ip}$</td>
</tr>
<tr>
<td>$\text{prn: } K$</td>
<td>$\text{sem: } \text{past}$</td>
</tr>
<tr>
<td>$\text{arg: } (\text{person } x) \ #\text{apple}$</td>
<td>$\text{arg: } (\text{person } y) \ #\text{apple}$</td>
</tr>
<tr>
<td>$\text{mdr:}$</td>
<td>$\text{mdr:}$</td>
</tr>
<tr>
<td>$\text{nc:}$</td>
<td>$\text{nc:}$</td>
</tr>
<tr>
<td>$\text{pc:}$</td>
<td>$\text{pc:}$</td>
</tr>
<tr>
<td>$\text{prn: } 33$</td>
<td>$\text{prn: } 33$</td>
</tr>
</tbody>
</table>

The grammatical function of the operations $N^S/V$ (subject gapping), $V^p/N$ (predicate gapping), and $N^O\setminus V$ (object gapping) is similar insofar as they manage the repeated transitions from the shared item to the first unmarked item on the gap list.
In contradistinction to N/V\((39)\), this operation has no instruction (the \textsc{fnc} value was already \#-marked in \[5.3.18\]).

The navigation has now traversed Bob bought an apple, Jim \(\emptyset\), and is currently at the shared predicate. Using the first unmarked item in its current gap list, here pear, it continues to the object of the gapped item:

### 5.3.23 Navigating with V\(\backslash\)N from buy to pear (arc 7)

\[\begin{align*}
\text{pattern level} & & \Rightarrow & & \text{content level} \\
\text{sur: lexnoun(\(\hat{\beta}\))} & & \Rightarrow & & \text{sur: a_pear} \\
\text{verb: buy} & & \Rightarrow & & \text{verb: buy} \\
\text{cat: #n' #a' decl +ip} & & \Rightarrow & & \text{cat: #n' #a' decl +ip} \\
\text{sem: past} & & \Rightarrow & & \text{sem: past} \\
\text{arg: #(person x) #apple} & & \Rightarrow & & \text{arg: #(person x) #apple} \\
\text{(#(person y) pear} & & \Rightarrow & & \text{(#(person y) pear} \\
\text{(person z) peach} & & \Rightarrow & & \text{(person z) peach} \\
\text{mdr:} & & \Rightarrow & & \text{mdr:} \\
\text{nc:} & & \Rightarrow & & \text{nc:} \\
\text{pc:} & & \Rightarrow & & \text{pc:} \\
\text{prn: 33} & & \Rightarrow & & \text{prn: 33}
\end{align*}\]

Using the \textsc{sem} values indef sg and the core value pear, lexnoun realizes the surface a_pear.

The navigation returns from pear to the shared predicate with N\(\backslash\)V to find the next unmarked item in the current gap list, here \((\text{person z})\):

### 5.3.24 Navigating with N\(\backslash\)V from pear to buy (arc 8)

\[\begin{align*}
\text{pattern level} & & \Rightarrow & & \text{content level} \\
\text{sur: lexverb(\(\hat{\alpha}\))} & & \Rightarrow & & \text{sur: lexverb(\(\hat{\alpha}\))} \\
\text{verb: \(\alpha\)} & & \Rightarrow & & \text{verb: \(\alpha\)} \\
\text{fnc: \(Z\)} & & \Rightarrow & & \text{fnc: \(\#b\)} \\
\text{mdr: \(\emptyset\)} & & \Rightarrow & & \text{mdr: \(\emptyset\)} \\
\text{prn: \(K\)} & & \Rightarrow & & \text{prn: \(K\)} \\
\text{Z is \#-marked or NIL} & & \Rightarrow & & \text{Z is \#-marked or NIL} \\
\text{sur: noun: pear} & & \Rightarrow & & \text{sur: noun: pear} \\
\text{cat: snp} & & \Rightarrow & & \text{cat: snp} \\
\text{sem: indef sg} & & \Rightarrow & & \text{sem: indef sg} \\
\text{fnc: #buy} & & \Rightarrow & & \text{fnc: #buy} \\
\text{mdr:} & & \Rightarrow & & \text{mdr:} \\
\text{nc:} & & \Rightarrow & & \text{nc:} \\
\text{pc:} & & \Rightarrow & & \text{pc:} \\
\text{prn: 33} & & \Rightarrow & & \text{prn: 33}
\end{align*}\]
The item pear in the gap list was #-marked as the result of the instruction in \( V \setminus N \) (5.3.22).

As the first unmarked item in the predicate’s current gap list, (person z) is the continuation value for \( V^p/N \):

### 5.3.25 Navigating with \( V^p/N \) from buy to bill (arc 9)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \text{sur: lexnoun(\( \beta \))} \\
\hline
\verb: \( \alpha \) & \text{noun: \( \beta \)} & \text{and_Bill} \\
\text{arg: \#X \( \beta \) Y} & \text{fnc: \( \alpha \)} & \text{cat: snp} \\
\text{prn: K} & \text{prn: K} & \text{sem: and nm m} \\
\hline
\text{sur: buy} & \text{cat: \#n’ \#a’ decl +ip} & \text{fnc: \#buy} \\
\text{verb: buy} & \text{sem: past} & \text{mdr:} \\
\text{cat: \#n’ \#a’ decl +ip} & \text{sem: past} & \text{nc:} \\
\text{arg: \#(person x) \#(person y) \#pear} & \text{arg: \#(person x) \#apple} & \text{pc:} \\
\text{(person z) peach} & \text{(person z) peach} & \text{prn: 33} \\
\text{mdr:} & \text{mdr:} & \text{prn: 33} \\
\text{nc:} & \text{nc:} & \text{pc:} \\
\text{pc:} & \text{pc:} & \text{prn: 33} \\
\text{prn: 33} & \text{prn: 33} & \\
\end{array}
\]

Lexnoun uses the name marker bill in the sur slot of bill and the initial sem value and to realize the surface and_Bill.

To find the object of the last gapped item, \( N/V^p \) returns to the shared predicate for a look at its current gap list:

### 5.3.26 Navigating with \( N/V^p \) from bill back to buy (arc 10)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \text{sur: lexverb(\( \alpha \))} \\
\hline
\text{noun: \( \beta \)} & \text{sur: buy} & \text{cat: \#n’ \#a’ decl +ip} \\
\text{fnc: \( \alpha \)} & \text{cat: \#n’ \#a’ decl +ip} & \text{sem: past} \\
\text{mdr: \( Z \)} & \text{arg: \#X \#\( \beta \) Y} & \text{arg: \#(person x) \#apple} \\
\text{prn: K} & \text{prn: K} & \text{arg: \#(person y) \#pear} \\
\text{Z is #-marked or NIL} & \text{mdr:} & \text{(person z) peach} \\
\text{sur: (person z)} & \text{mdr:} & \text{nc:} \\
\text{cat: snp} & \text{mdr:} & \text{pc:} \\
\text{sem: and nm m} & \text{mdr:} & \text{prn: 33} \\
\text{fnc: \#buy} & \text{mdr:} & \text{prn: 33} \\
\text{mdr:} & \text{nc:} & \text{prn: 33} \\
\text{nc:} & \text{pc:} & \text{prn: 33} \\
\text{pc:} & \text{prn: 33} & \\
\text{prn: 33} & \text{prn: 33} & \\
\end{array}
\]
As in 5.3.21, lexverb does not realize any surface because the second arg value apple of buy is #-marked (5.3.19).

Using the last unmarked value in the current gap list, V\N applies as follows:

### 5.3.27 Navigating with V\N from buy to peach (arc 11)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
<th>Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: α</td>
<td>sur: lexnoun(β)</td>
<td>#-mark β in the arg slot of proplet α</td>
</tr>
<tr>
<td>arg: #X β Y</td>
<td>noun: β</td>
<td></td>
</tr>
<tr>
<td>prn: K</td>
<td>fnc: α</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mdr: Z</td>
<td></td>
</tr>
<tr>
<td></td>
<td>nc:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pc:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>prn: 33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sur: buy</td>
<td>verb: α</td>
</tr>
<tr>
<td></td>
<td>cat: #n′ #a′ decl +ip</td>
<td>arg: #(person x) #apple</td>
</tr>
<tr>
<td></td>
<td>sem: past</td>
<td>#m: #p</td>
</tr>
<tr>
<td></td>
<td>arg: #m: #p</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mdr: #buy</td>
<td>fn: #buy</td>
</tr>
<tr>
<td></td>
<td>nc:</td>
<td>mdr:</td>
</tr>
<tr>
<td></td>
<td>pc:</td>
<td>pc:</td>
</tr>
<tr>
<td></td>
<td>prn: 33</td>
<td>prn: 33</td>
</tr>
</tbody>
</table>

It remains to return to the shared predicate to realize the period and to reach a location potentially suitable for navigating along an extrapropositional coordination to a next proposition (1.4.4):

### 5.3.28 Navigating with N\V from peach to buy (arc 12)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Content Level</th>
<th>Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: β</td>
<td>sur: lexverb(α)</td>
<td></td>
</tr>
<tr>
<td>fnc: α</td>
<td>verb: α</td>
<td></td>
</tr>
<tr>
<td>mdr: Z</td>
<td>arg: #X β Y</td>
<td></td>
</tr>
<tr>
<td>prn: K</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>sur: peach</td>
<td>verb: buy</td>
</tr>
<tr>
<td></td>
<td>cat: snp</td>
<td>arg: #n′ #a′ decl +ip</td>
</tr>
<tr>
<td></td>
<td>sem: indef sg</td>
<td>sem: past</td>
</tr>
<tr>
<td></td>
<td>fn: #buy</td>
<td>arg: #(person x) #apple</td>
</tr>
<tr>
<td></td>
<td>mdr:</td>
<td>#m: #p</td>
</tr>
<tr>
<td></td>
<td>nc:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pc:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>prn: 33</td>
<td>prn: 33</td>
</tr>
</tbody>
</table>

Extending DBS.20 to an example class with predicate gapping required the definition of the hear mode operation PRD^P × SBJ (t30) and of the speak mode operations V^P/N, (s25), N/V^P (s26), and N/V^i (s39). The hear and the speak mode derivation used 12 operation applications each.
5.3.29 Comparing Hear and Speak Mode Surface Segmentation

**Hear Mode:**
1  2  3  4  5  6  7
Bob \(\times\) bought \(\times\) an \(\cup\) apple \(\times\) Jim \(\times\) a \(\cup\) pear \(\times\)
8a  8b  9  10  11
and \(\cup\) \(\times\) Bill \(\times\) a \(\cup\) peach \(\cup\).

**Speak Mode:**
1  2  3  4  5  6
VP\(\uparrow\)/N Bob N\(\uparrow\)/VP\(\downarrow\) bought V\(\downarrow\)N an_apple N\(\uparrow\)\V\(\downarrow\)VP\(\downarrow\)/N Jim N\(\uparrow\)/VP\(\downarrow\)
7  8  9  10  11  12
V\(\downarrow\)N a_pear_and N\(\uparrow\)\V\(\downarrow\)VP\(\downarrow\)/N Bill N\(\uparrow\)/VP\(\downarrow\)\V\(\downarrow\)N a_peach N\(\uparrow\)/V.

The breadth-first numbering of the speak mode operations corresponds to the arc traversal numbers of the NAG in [5.3.14].

5.4 DBS.22: Repeating Object Gapping

Pretheoretically, object gapping may be illustrated as follows:

Bob buy \(\emptyset\)
Jim peel \(\emptyset\)
Bill eat peach

Compared to subject and predicate gapping, the gaps in object gapping precede the shared item, i.e. the grammatical object. Therefore the time-linear hear mode derivation must collect the verbs of the gapped items in a storage variable \(v_o\), to be inserted as the gap list into the fnc slot of the shared object peach when it finally arrives.

In the content, however, these procedural circumstances of the time-linear derivation are not reflected, as shown by the following example:

5.4.1 Bob bought \(\emptyset\), Jim peeled \(\emptyset\), and Bill ate a peach.
The gap list is the same as in the subject gapping example 5.2.1, but it is stored in the object proplet (and not the subject). The verb proplets in this order-free set take the core value peach of the shared object as their non-initial arg value.

Unlike subject (Sect 5.2) and predicate gapping (Sect 5.3), the construction of object gapping does not begin with a complete sentence. Instead the complete sentence containing the shared object comes at the end. Consider the time-linear surface compositional hear mode derivation in DBS graph format:

**5.4.2 Graphical hear mode derivation of the content 5.4.1**

storage variables were first used in Montague grammar (Cooper 1983). The storage variable \( v_o \) used here is non-recursive and its scope is an elementary proposition (because gapping is intrapositional).
The derivation steps of line 2 and 5b write to the storage variable with \( \text{buy} \rightarrow v\_o \) and \( \text{peel} \rightarrow v\_o \). They also write the substitution variable \( n\_c \) (noun constant) into the second (i.e. object) arg slot of \( \text{buy} \) and \( \text{peel} \), respectively.

The derivation step of line 7 moves the content of the storage variable \( v\_o \)
5. Unbounded Repetition

into the \text{n}c slot of the (beginning of the) shared object \textit{peach} \text{(release \text{v}._o content), resulting in the gap list \textit{buy peel}. The verb of the last gapped item, i.e. \textit{eat}, is copied directly to the end of the gap list, now in the \textit{peach} proplet [5.4.10]. The substitution variable \text{n}_c is replaced with \text{n}_2 as the core value of (the beginning of) the shared object; in this way all verb proplets mentioned in the gap list are provided with the address of the shared item (object).

Following standard procedure, let us turn to point 3 on the to-do list 1.5.1:

5.4.3 \textbf{Cross-copying \textit{Bob}} and \textbf{bought} with \textbf{SBJ} \times \textbf{PRD} (line 1)

\begin{align*}
\text{SBJ} \times \text{PRD} & \quad \text{(1)} \\
\text{pattern level} & \\
\begin{array}{|c|c|c|}
\hline
\text{noun: } \alpha & \text{verb: } \beta & \text{noun: } \alpha \\
\text{cat: NP} & \text{cat: NP'} X v & \text{cat: NP} \\
\text{fnc:} & \text{arg:} & \text{fnc:} \\
\text{prn: K} & \text{prn:} & \text{prn: K} \\
\hline
\end{array} \\
\text{content level} & \\
\begin{array}{|c|c|c|}
\hline
\text{sur: Bob} & \text{sur: bought} & \text{sur: bob} \\
\text{noun: (person x)} & \text{verb: buy} & \text{noun: (person x)} \\
\text{cat: snp} & \text{cat: n' a' v} & \text{cat: snp} \\
\text{sem: nm m} & \text{sem: past} & \text{sem: nm m} \\
\text{fnc:} & \text{arg:} & \text{fnc:} \\
\text{nc:} & \text{mdr:} & \text{nc:} \\
\text{pc:} & \text{pm:} & \text{pc:} \\
\text{prn: 34} & \text{prn:} & \text{prn: 34} \\
\hline
\end{array}
\end{align*}

The next proplet provided by automatic word form recognition is \textit{jim}. One interpretation is as the grammatical object of a simple declarative. The other is combining it with the following predicate \textit{buy} of the first gapped item with the new operation \textbf{PRD} \times \textbf{SBJ}, with \textbf{0} showing the shared item to be the object. The first reading is excluded by the arrival of \textit{buy}.

5.4.4 \textbf{Adding \textit{buy} to the gap list on \textit{v}_o with PRD} \times \textbf{SBJ} (line 2)

\begin{align*}
\text{PRD} \times \text{SBJ} & \quad \text{(51)} \\
\text{pattern level} & \\
\begin{array}{|c|c|c|}
\hline
\text{verb: } \alpha & \text{noun: } \beta & \text{verb: } \alpha \\
\text{cat: } #X' \ N' v & \text{cat: } #X' \ N' v & \text{cat: } #X' \ N' v \\
\text{arg: } Y & \text{arg: } Y \ n_c & \text{arg: } Y \ n_c \\
\text{prn: K} & \text{prn: K} & \text{prn: K} \\
\hline
\end{array} \\
\text{content level} & \\
\begin{array}{|c|c|c|}
\hline
\text{sur: Jim} & \text{sur: buy} & \text{sur: jim} \\
\text{noun: (person y)} & \text{cat: snp} & \text{noun: (person y)} \\
\text{cat: } #n' \ a' v & \text{cat: #n'} & \text{cat: snp} \\
\text{sem: past} & \text{sem: past} & \text{sem: past} \\
\text{arg: (person x)} & \text{arg: (person x) } n_c & \text{arg: (person x) } n_c \\
\text{mdr:} & \text{mdr:} & \text{mdr:} \\
\text{nc:} & \text{nc:} & \text{nc:} \\
\text{pc:} & \text{pc:} & \text{pc:} \\
\text{prn: 34} & \text{prn: 34} & \text{prn: 34} \\
\hline
\end{array}
\end{align*}
The first output pattern provides the arg slot of buy with n_c as the place holder for the object valency position, with concomitant #-marking of the oblique valency position a’ in the cat slot. The instruction α → v_o copies the core value of buy to the storage variable v_o.

When automatic word form recognition provides peel, standard SBJ×PRD combines the subject and the predicate of the second gapped item:

5.4.5 Cross-copying Jim and peeled with SBJ×PRD (line 3)

The hear mode derivation has now processed the input Bob bought | Jim peeled, with buy stored in the v_o variable and peel to be added in 5.4.8.

The conjunction and is suspended until Bill arrives:

5.4.6 Suspending peel and and with V~CNJ (line 4)

---

* If these properties are ignored by the system, there is a brief ambiguity between the interpretation of Jim as an object vs. as the subject of a gapped item, which is resolved with the arrival of peel in 5.4.3.
The next word proplet \textit{bill} preserves the kind of the conjunction by storing its English surface, here \textit{and}, in its initial \texttt{sem} slot:

### 5.4.7 Absorbing $n_1$ into Bill with \texttt{CNJ}$

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{CNJ} $\cup$ \texttt{N} \texttt{(5.3)}</td>
<td>\texttt{surf: Bill} \texttt{noun: (person z)} \texttt{cat: snp} \texttt{sem: nm m} \texttt{fnc: mdr: nc: pc: prn: 34}</td>
</tr>
<tr>
<td>\texttt{noun: $n_1$} \texttt{sem: $\beta$} \texttt{prn: K} \texttt{$\beta$ is (and, or)}</td>
<td>\texttt{noun: $\alpha$} \texttt{sem: X} \texttt{prn: }</td>
</tr>
</tbody>
</table>

At the now front, the input to the operation is replaced by its output. As in [3.6.5] [5.2.10] and [5.3.10] no \texttt{prn} value is assigned to the output.

The next operation to apply is \texttt{PRD}$\times$\texttt{SBJ} \texttt{(5.4.4)}. Except for the value \texttt{and} in the \texttt{sem} slot of \texttt{bill}, this hear mode derivation step could result in the beginning of a third gapped item. In fact, however, the addition of \texttt{bill} turns out to be the beginning of the complete final sentence:

### 5.4.8 Adding peel with \texttt{PRD}$\times$\texttt{SBJ} to the gap list of \texttt{v_o} \texttt{(line 5b)}

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{PRD}$\times$\texttt{SBJ} \texttt{(5.3)}</td>
<td>\texttt{surf: bill} \texttt{noun: (person y)} \texttt{cat: snp} \texttt{sem: and nm m} \texttt{fnc: mdr: nc: pc: prn: 34}</td>
</tr>
<tr>
<td>\texttt{verb: $\alpha$} \texttt{cat: #$n'$ #a' v} \texttt{arg: (person y)} \texttt{prn: K}</td>
<td>\texttt{noun: $\beta$} \texttt{cat: #$n'$ #n v} \texttt{arg: Y n_c} \texttt{prn: K}</td>
</tr>
</tbody>
</table>

The place holder \texttt{n_c} is placed into the second \texttt{arg} slot of the predicate, the valency position \texttt{a'} is \#-marked, and \texttt{peel} is added to the gap list of \texttt{v_o} (written as the instruction $\alpha \rightarrow \texttt{v_o}$).

The hear mode derivation has now processed the input \textit{Bob bought $\emptyset$ Jim peeled $\emptyset$, and Bill, with \textit{buy} and \textit{peel} stored in \texttt{v_o}. It remains to complete
the final sentence. The first step is combining the subject and the newly arrived predicate with standard SBJ×PRD, similar to 5.4.3 and 5.4.5.

### 5.4.9 Cross-copying Bill and *ate* with SBJ×PRD (line 6)

The special tasks of this operation are to (i) move the gap list from \( v_o \) into the \( \text{fnc} \) slot of the shared object, written as \( V_o \), (ii) add the last verb, here *eat* bound to \( \beta \), to the gap list, and (iii) to provide the verb proplets named by the gap list with the core value of the shared object, here *peach*, as their non-initial arg value:

### 5.4.10 Cross-copying *ate* and *a(n)* with PRD\(^0\)×OBJ (line 7)

The new operation PRD\(^0\)×OBJ applies. The special tasks of this operation are to (i) move the gap list from \( v_o \) into the \( \text{fnc} \) slot of the shared object, written as \( V_o \), (ii) add the last verb, here *eat* bound to \( \beta \), to the gap list, and (iii) to provide the verb proplets named by the gap list with the core value of the shared object, here *peach*, as their non-initial arg value:

---

**Table:**

<table>
<thead>
<tr>
<th>SBJ×PRD (41)</th>
<th>PRD(^0)×OBJ (27)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pattern level</strong></td>
<td><strong>Pattern level</strong></td>
</tr>
<tr>
<td>[noun: ( \alpha )] cat: NP</td>
<td>[verb: ( \beta )] cat: #X(^\prime) N( ^\prime ) v arg: Z</td>
</tr>
<tr>
<td>fnc:</td>
<td>fnc:</td>
</tr>
<tr>
<td>prn: K</td>
<td>prn: K</td>
</tr>
</tbody>
</table>

**Content level**

<table>
<thead>
<tr>
<th>[sur: bill]</th>
<th>[sur: ( \alpha )]</th>
<th>[sur: bill]</th>
<th>[sur: ( \alpha )]</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: (person z)</td>
<td>noun: ( \alpha )</td>
<td>noun: (person z)</td>
<td>noun: ( \alpha )</td>
</tr>
<tr>
<td>cat: snp</td>
<td>cat: #X(^\prime) N( ^\prime )</td>
<td>cat: snp</td>
<td>cat: #X(^\prime) N( ^\prime )</td>
</tr>
<tr>
<td>fnc: and nm m</td>
<td>fnc:</td>
<td>fnc:</td>
<td>fnc:</td>
</tr>
<tr>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
<td>mdr:</td>
</tr>
<tr>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
<td>pc:</td>
</tr>
<tr>
<td>prn: 34</td>
<td>prn:</td>
<td>prn:</td>
<td>prn:</td>
</tr>
</tbody>
</table>

**Agreement conditions as in 2.1.2.**

Next automatic word form recognition provides the object proplet, beginning with \( a(n) \). For connecting it with the predicate *eat*, the standard PRD×OBJ operation h\(^{22}\) can not be used because the object functions as the shared item of the gapping construction.

Instead the new operation PRD\(^0\)×OBJ applies. The special tasks of this operation are to (i) move the gap list from \( v_o \) into the \( \text{fnc} \) slot of the shared object, written as \( V_o \), (ii) add the last verb, here *eat* bound to \( \beta \), to the gap list, and (iii) to provide the verb proplets named by the gap list with the core value of the shared object, here *peach*, as their non-initial arg value:

| Replace all occurrences of \( n_o \) with \( a(n) \) by simultaneous substitution. |
The tasks (i) and (ii) are coded by the feature $[\text{fnc: } V_\beta]$ in the second output pattern: $V_\beta$ tells the storage variable $v_o$ to release its content into the object proplet and $\beta$ adds the final verb. Task (iii) is specified by the instruction.

Next automatic word form recognition provides the common noun peach. It is absorbed into the determiner by standard $\text{DET} \cup \text{CN}$, which replaces all occurrences of $n_2$ with peach by simultaneous substitution.

5.4.11 Absorbing peach into a(n) with $\text{DET} \cup \text{CN}$ (line 8)

At this point, all verbs have the value peach in their second (object) arg slot.

The final step of the hear mode derivation is adding the period with $\text{S} \cup \text{IP}$.

5.4.12 Absorbing • into buy with $\text{S} \cup \text{IP}$ (line 9)

As in subject gapping (5.2.14), the most recent verb is the most natural for continuing to a next proposition.

Following standard procedure let us turn to point 4 on the to-do list [1.5.1], i.e. the canonical DBS graph analysis for the speak mode.
5.4.13 **Graph analysis underlying production of**

(i) **SRG (semantic relations graph)**

![SRG diagram](image)

(ii) **signature**

![Signature diagram](image)

(iii) **NAG (numbered arcs graph)**

![NAG diagram](image)

(iv) **surface realization**

<table>
<thead>
<tr>
<th>Arc</th>
<th>Source</th>
<th>Target</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V/N</td>
<td>from buy to bob</td>
<td>Bob</td>
</tr>
<tr>
<td>2</td>
<td>N/V</td>
<td>from bob to buy</td>
<td>bought</td>
</tr>
<tr>
<td>3</td>
<td>V\N⁰</td>
<td>from buy to peach</td>
<td>peeled</td>
</tr>
<tr>
<td>4</td>
<td>V/N</td>
<td>from peel to jim</td>
<td>Jim</td>
</tr>
<tr>
<td>5</td>
<td>N/V</td>
<td>from jim to peel</td>
<td>peeled</td>
</tr>
<tr>
<td>6</td>
<td>V\N⁰</td>
<td>from peel to peach</td>
<td>peeled</td>
</tr>
<tr>
<td>11</td>
<td>N⁰\V</td>
<td>from peach to eat</td>
<td>and_Bill</td>
</tr>
<tr>
<td>8</td>
<td>V/N</td>
<td>from eat to bill</td>
<td>and_Bill</td>
</tr>
<tr>
<td>9</td>
<td>N/V</td>
<td>from bill to eat</td>
<td>ate</td>
</tr>
<tr>
<td>10</td>
<td>V\N⁰</td>
<td>from eat to peach</td>
<td>a_peach</td>
</tr>
<tr>
<td>11</td>
<td>N⁰\V</td>
<td>from peach to eat</td>
<td>.</td>
</tr>
</tbody>
</table>

The shared object is clearly shown. Just as the graph for subject gapping is missing a downward arc opposite arc 9, the current graph for object gapping is missing an upward arc opposite arc 3. The different tilts of the three N/V lines are solely for visual separation in the graph.

We continue with point 5 on the to-do list i.e. the list of speak mode operation names with surface realizations:

5.4.14 **Sequence of operation names and surface realizations**

9 Because it is PRD⁰ \times SBJ 5.4.4, 5.4.8, which adds the n_c value to the second arg slot of the previous gapped item’s predicate (the last time in line 5b of 5.4.2), there has been no occasion to provide n_c to the second arg slot of final eat. Instead, n_2 is copied there directly 5.4.2, line 7).
The traversals take the shortest route, i.e. from peach via peel to Jim and from the shared object peach via eat to Bill, which is why arc 3 has no upward counterpart.

The complete sequence of explicit speak mode operation applications begins with V/N traversing arc 1:

5.4.15 Navigating with V/N from buy to bob (arc 1)

\[
\begin{array}{l}
\text{pattern level} \quad \text{content level}
\end{array}
\]

\[
\begin{array}{l}
\text{verb: } \alpha \quad \text{sur: lexnoun(} \beta \text{)}
\end{array}
\]

\[
\begin{array}{l}
\text{arg: } \beta X
\end{array}
\]

\[
\begin{array}{l}
\text{prn: } K
\end{array}
\]

\[
\begin{array}{l}
\text{noun: } \beta
\end{array}
\]

\[
\begin{array}{l}
\text{fnc: } \alpha
\end{array}
\]

\[
\begin{array}{l}
\text{prn: } K
\end{array}
\]

\[
\begin{array}{l}
\text{#-mark } \beta \text{ in the arg slot of proplet } \alpha
\end{array}
\]

Because neither the input nor the output proplet is a shared item (no gap list) standard V/N may be used.

Next, N/V\textsuperscript{10} travels from bob back to buy and realizes bought:

5.4.16 Navigating with N/V from bob back to buy (arc 2)

\[
\begin{array}{l}
\text{pattern level} \quad \text{content level}
\end{array}
\]

\[
\begin{array}{l}
\text{noun: } \beta
\end{array}
\]

\[
\begin{array}{l}
\text{fnc: } \alpha
\end{array}
\]

\[
\begin{array}{l}
\text{mdr: } Z
\end{array}
\]

\[
\begin{array}{l}
\text{prn: } K
\end{array}
\]

\[
\begin{array}{l}
\text{sur: lexverb(} \alpha \text{)}
\end{array}
\]

\[
\begin{array}{l}
\text{verb: } \alpha
\end{array}
\]

\[
\begin{array}{l}
\text{arg: } #\beta X
\end{array}
\]

\[
\begin{array}{l}
\text{prn: } K
\end{array}
\]

\[
\begin{array}{l}
\text{#-mark } \alpha \text{ in the fnc slot of proplet } \beta
\end{array}
\]

\[
\begin{array}{l}
\text{Z is NIL, or elementary and #-marked}
\end{array}
\]

\[
\begin{array}{l}
\text{sur: (person x)}
\end{array}
\]

\[
\begin{array}{l}
\text{cat: snp}
\end{array}
\]

\[
\begin{array}{l}
\text{sem: nm m}
\end{array}
\]

\[
\begin{array}{l}
\text{fnc: buy}
\end{array}
\]

\[
\begin{array}{l}
\text{mdr: }
\end{array}
\]

\[
\begin{array}{l}
\text{nc: }
\end{array}
\]

\[
\begin{array}{l}
\text{pc: }
\end{array}
\]

\[
\begin{array}{l}
\text{prn: 34}
\end{array}
\]

\[
\begin{array}{l}
\text{sur: bought}
\end{array}
\]

\[
\begin{array}{l}
\text{verb: buy}
\end{array}
\]

\[
\begin{array}{l}
\text{cat: #n’ #a’ decl}
\end{array}
\]

\[
\begin{array}{l}
\text{sem: past}
\end{array}
\]

\[
\begin{array}{l}
\text{arg: #(person x) peach}
\end{array}
\]

\[
\begin{array}{l}
\text{mdr: }
\end{array}
\]

\[
\begin{array}{l}
\text{nc: }
\end{array}
\]

\[
\begin{array}{l}
\text{pc: }
\end{array}
\]

\[
\begin{array}{l}
\text{prn: 34}
\end{array}
\]

At this point, it is open for the hearer, whether Bob bought will be continued into a complete declarative sentence or with another gapped item.

\textsuperscript{10} In contradistinction to N\textsuperscript{6}/V (subject gapping, 24), N/V is standard (no superscript, 6.4.6).
To get to the beginning of the second gapped item, i.e. \( \text{person y} \) for Jim, the navigation first moves to the shared object to get the second item on the gap list, i.e. \text{peel}, as a means to find the second subject. This is based on the new operation \( V \setminus N^0 \):

5.4.17 Navigating with \( V \setminus N^0 \) from \text{buy} to \text{peach} (arc 3)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \quad \text{content level} \\
\verb: \alpha & \quad \text{arg: } \#X \beta \ Y \\
\text{prn: } K & \quad \text{sur: lexnoun}(\hat{\beta}) \\
\end{align*}
\]

\[
\begin{align*}
\text{sur: } & \quad \text{noun: } \beta \ Y \alpha \ Z \\
\text{fnc: } & \quad \text{prn: } K \\
\end{align*}
\]

\[
\begin{align*}
\text{verb: } & \quad \text{arg: } \#X \beta \ Y \\
\text{prn: } & \quad \text{sur: lexverb}(\hat{\alpha}) \\
\end{align*}
\]

\[
\begin{align*}
\text{arg: } & \quad \text{cat: } \#n' \ #a' \ v \\
\text{sem: } & \quad \text{sem: } \text{past} \\
\text{fnc: } & \quad \text{arg: } \text{(person y) peach} \\
\text{peel } & \quad \text{mdr: } \\
\text{eat } & \quad \text{nc: } \\
\text{pc: } & \quad \text{prn: } 34 \\
\end{align*}
\]

The result of the instruction shows up in the fnc slot of the start proplet in 5.4.18. Any unmarked prefinal value on the gap list prevents lexnoun from realizing the surface \text{a peach} of the shared object. Thus the traversal to the object noun is empty, but provides the next unmarked item on the gap list of \text{peach}, here \text{peel}.

The traversal from the shared object to \text{peel} is based on new \( N^0 \setminus V \).

5.4.18 Navigating with \( N^0 \setminus V \) from \text{peach} to \text{peel} (arc 7)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \quad \text{content level} \\
\text{noun: } & \quad \text{sur: } \text{lexverb}(\hat{\alpha}) \\
\beta & \quad \text{verb: } \alpha \\
\text{fnc: } & \quad \text{arg: } \beta \ W \\
\text{mdr: } & \quad \text{prn: } K \\
\text{prn: } K & \quad Z \text{ is } \#\text{-marked or NIL} \\
\end{align*}
\]

\[
\begin{align*}
\text{sur: } & \quad \text{noun: peach} \\
\text{cat: } & \quad \text{cat: } \#n' \ #a' \ v \\
\text{sem: } & \quad \text{sem: } \text{past} \\
\text{fnc: } & \quad \text{arg: } \text{(person y) peach} \\
\text{peel } & \quad \text{mdr: } \\
\text{eat } & \quad \text{nc: } \\
\text{pc: } & \quad \text{prn: } 34 \\
\end{align*}
\]
At this point, the navigation has realized Bob bought ∅ and traversed the proplets buy, bob, buy, peach, peel.

Standard V/N provides the traversal from peel to the subject jim of the second gapped item. Analogous to 5.4.15 the surface Jim is realized:

5.4.19 NAVIGATING WITH V/N FROM peel TO jim (arc 4)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \\
\text{content level} & \\
\end{align*}
\]

The following return to peel with standard N/V realizes the surface of the second gapped item’s predicate:

5.4.20 NAVIGATING WITH N/V FROM jim TO peel (arc 5)

\[
\begin{align*}
\text{pattern level} & \quad \Rightarrow \\
\text{content level} & \\
\end{align*}
\]

The navigation has now traversed buy, bob, buy, peach, peel, jim, peel and realized the surfaces Bob bought ∅ Jim peeled.

To get to the beginning of the final sentence, i.e. the subject (person z) for Bill, the navigation first returns to the shared object to get the third predicate on the gap list as a means to access the third subject. This is based on another application of V\N^o (5.4.17):
5.4.21 Navigating with $V \setminus N^0$ from peel to peach (arc 6)

As in [5.4.17], the instruction applies to a value in the goal proplet: it $#-marks the second item on the gap list. The result is shown in the $fnc$ slot of the start proplet when the next operation $N^0 \setminus V$ reapply (5.4.18):

5.4.22 Navigating with $N^0 \setminus V$ from peach to eat (arc 11)

The pattern $[fnc: #X α Y]$ of the start pattern binds the values $#buy #peel$ to $#X$, the value $eat$ to $α$, and the value NIL to $Y$. The reoccurrence of $α$ in the goal pattern enables access to the predicate $eat$ of the final sentence.

From there, the navigation moves with $V/\mathbf{N}$ to the subject and realizes and_Bill with lexnoun:
5.4.23 Navigating with V/N from eat to bill (arc 8)

\[ \begin{align*}
\text{pattern level} & \\
\verb: \alpha & \\
\text{arg: } \beta \& \ X & \\
\text{prn: } K & \\
\rightarrow & \\
\text{content level} & \\
\text{sur: } \langle \text{lexnoun}(\beta) \rangle & \\
\text{noun: } \beta & \\
\text{fnc: } \alpha & \\
\text{prn: } K & \\
\end{align*} \]

Lexnoun uses the initial sem value and the name marker in the sur slot of the goal proplet to realize and_Bill.

From the subject bill of the final sentence, the navigation returns with standard N/V to the predicate and realizes ate:

5.4.24 Navigating with N/V from bill to eat (arc 9)

\[ \begin{align*}
\text{pattern level} & \\
\text{noun: } \beta & \\
\text{fnc: } \alpha & \\
\text{mdr: } Z & \\
\text{prn: } K & \\
\rightarrow & \\
\text{content level} & \\
\text{sur: } \langle \text{lexverb}(\alpha) \rangle & \\
\text{verb: } \alpha & \\
\text{arg: } #\beta \& \ X & \\
\text{prn: } K & \\
\end{align*} \]

Using the second arg value peach of eat as the continuation value, the navigation proceeds with V\textbackslash N^f to the shared object.

5.4.25 Navigating with V\textbackslash N^f from eat to peach (arc 10)

\[ \begin{align*}
\text{pattern level} & \\
\verb: \alpha & \\
\text{arg: } #\beta \& \ X & \\
\text{prn: } K & \\
\rightarrow & \\
\text{content level} & \\
\text{sur: } \langle \text{lexnoun}(\beta) \rangle & \\
\text{noun: } \beta & \\
\text{fnc: } #\alpha & \\
\text{prn: } K & \\
\end{align*} \]
V\Nf differs from V\No (s29) because it has two #-marking instructions. The results of the instructions show up in the following operation application. With all items but one on the gap list #-marked, lexnoun realizes the shared object a_peach.

Using the last fnc value eat, now #-marked, of the shared object peach as the continuation value, the navigation proceeds with the last predicate to realize the period. Nf\V differs from No \V (s30) because it requires all items on the gap list to be #-marked.

5.4.26 Navigating with Nf\V from peach to buy (arc 11)

With all continuation values in the eat proplet #-marked, lexverb realizes the period, using the cat value decl.

Following standard procedure, we conclude with point 7 on the to-do list Extending DBS.21 to an example class with object gapping required the definition of two hear mode operations, PRD^9 \times SBJ (h31) and PRD^9 \times OBJ (h27), and four speak mode operations, V\Nf (s27), Nf\V (s28), V\No (s29),
and \( N^0 \backslash V \) \( (30) \). The hear mode derivation used ten operation applications and the speak mode 12.

5.4.27 Comparing Hear and Speak Mode Surface Segmentation

Hear mode:

1 2 3 4 5a 5b 6 7
Bob \( \times \) bought \( \rightarrow^{11} \) Jim \( \times \) peeled \( \sim \) and \( \cup \rightarrow \) Bill \( \times \) ate \( \times \)
8 9
\( a \cup \) peach \( \cup \).

Speak mode:

1 2 3 7 4 5
\( V/\)ssearrow \( N \) Bob \( N/\)nearrow \( V \) bought \( V/\)N \( \) N\( N^0 \backslash V \) V\( N/\)N Jim \( N/\)V peeled
6 11 8 9 10 11
\( V/\)N\( N^0 \) and \( N^0 \backslash V \) V\( N/\)N \( \) Bill \( N/\)V ate \( V/\)N\( \) \( a \)\_peach \( N^1 \backslash V \) .

The numbering of the speak mode operations corresponds to the arc traversal numbers of the NAG in 5.4.13. As in subject gapping (5.2.28), there is one multiple traversal, here arc 11.

5.5 DBS.23: Repeating Object Clauses

While repeating prepositional phrases (Sect. 5.1) and subject (Sect. 5.2), predicate (Sect. 5.3), and object gapping (Sect. 5.4) are intrapropositional constructions, repeating object clauses with an initial \( wh \)-interrogative (aka unbounded dependency) are extrapropositional, just like a simple clausal object construction (Sect. 2.6). We begin the analysis with point 1 on the to-do list 1.5.1, i.e. the example surface and the set of proplets representing its content:

5.5.1 Whom does John say that Bill believes that Mary loves?

11 The \( \rightarrow \) indicates writing to the storage variable.
The extrapropositional nature of the construction is formally reflected by the use of several prn values.

In accordance with point 2 of the DBS laboratory set-up 1.5.1, let us turn next to the time-linear hear mode derivation in canonical DBS graph format:

5.5.2 Graphical hear mode derivation of the content 5.5.1

The declarative variant John said that Bill believes that Mary loves Tom. has the same semantic relations except that the final object is Tom rather than initial whom (NLC Sect. 7.6).
In DBS, an unbounded dependency has the form of an unbounded suspension, here between whom in line 1 and the final transitive verb form loves in line 9b. Because the length of the suspension is only determined at the end of the input sentence, the time-linear surface compositional hear mode derivation is subject to the following structure of syntactic-semantic ambiguity (NLC 7.6.5):
5.5.3 AMBIGUITY STRUCTURE OF AN UNBOUNDED SUSPENSION

A. Whom does John love?

B. does John say that Bill loves?

C. that Bill believes that Mary loves?

D. that Mary claims that Suzy loves?

Whom must belong to a proposition with a two-place (transitive) or three-
place (ditransitive) verb, because it is marked morphologically as oblique.

Viewing the sentence in isolation, the prn value of the WH proplet is clear:
it can only be 1 because whom is sentence initial. The ambiguity is whether
the prn value of the next word, here does, should (i) stay in the same (line A)
or (ii) be incremented (line B).

In line A, whom is the object of an elementary proposition with a transitive
verb which does not take a clausal object. Thus all proplets in line A share the
prn value 1. In line B, in contrast, the matrix verb takes a clausal object which
whom belongs to. Thus, does John say X has the prn value 2, whereby X
is Bill loves whom with the prn value 1. The construction in line B must
terminate because the verb love does not take a clausal object.

Line C branches off line B because the first object clause uses a verb which
takes a second object clause as its oblique argument. Thus, does John say X
continues to have the prn value 2, but the new object clause Bill believes Y
has the new prn value 3, whereby Y is that Mary loves whom with the prn
value 1. The construction terminates in branch C.

Line D branches off C because the second object clause uses a verb which
takes a third object clause as its argument. Thus, does John say X continues
to have the prn value 2, Bill believes Y continues to have the prn value 3, but
the new object clause that Mary claims Z has the prn value 4, whereby Z is
that Suzy loves whom with the prn value 1, etc.

Even though an unbounded suspension (i) may be continued indefinitely and
(ii) causes a systematic syntactic ambiguity, it does not increase the computa-
tional complexity of natural language (FoCL Sect. 11.5). This is because
one of the two branches terminates when the next ambiguity begins. In other
words, there is no single return ambiguity in 5.5.3 unlike a garden path sen-
tence (FoCL 11.3.6).

For simplicity, the following sequence of operations will be “deterministic”
in the sense of classic computational complexity theory (FoCL Sect. 12.1).

13 For the distinction between syntactic, semantic, and pragmatic ambiguity see FoCL Sects. 11.3, 12.5.
Accordingly, each derivation step will show only those operation applications which lead to the intended result; branches which are caused by the ambiguity structure of the example but turn out to be dead ends are omitted.

The hear mode derivation begins with the new hear mode operation WH~AUX. In accordance with the overall interpretation of the example, the operation suspends WH and AUX, but increments the prn value of the AUX proplet. The prn value of the WH proplet is marked with & as initial:

5.5.4 SUSPENDING Whom and does with WH~AUX (line 1)

The &-marker will be used by PRD_{sub} thatke (5.5.11) to assign the proper prn value when it enters the final subclause, which initial whom belongs to semantically.

Next, the finite auxiliary and the subject of the second proposition are combined with the new operation do\times NOM:

5.5.5 CROSS-COPYING \_1 and John with do\times NOM (line 2)
Compared to SBJ×PRD, the order of the noun and verb patterns is inverted in do×NOM, but the agreement conditions are the same.

The next operation absorbs the infinitive say into the auxiliary do and changes the category segment v to vi, enforcing a question mark at the end.

5.5.6 Absorbing say into does with do∪INF (line 3)

For the English infinitive, DBS uses the unmarked present tense form of the main verb matching the pattern [cat: n-s3’ X v] (CLAqTR Sects. 15.4–15.6). The result of the absorption uses the nominative valency of the finite auxiliary.

The first object sentence begins in line 4 of 5.5.2 with the arrival of the conjunction that, represented by its core value v_2. This derivation step corresponds to line 2 in 2.6.4 and is based on the same operation (2.6.4):

5.5.7 Cross-copying say and that with PRD×subthat (line 4)

The lexical entry specifies the kind of conjunction by storing its English surface, here that, in the initial sem slot of the next word proplet. It has the core
attribute verb and the continuation attributes (i) arg for the subclause and (ii) fnc for the connection to the higher clause. The operation connects say and the (beginning of the) first object clause by copying the core value \( v_2 \) into the second arg slot of say and say into the fnc slot of \( v_2 \).

The next three derivation steps \[5.5.8, 5.5.10\] are analogous to \[2.6.5, 2.6.7\].

5.5.8 SUSPENDING that AND Bill WITH subclause~SBJ (line 5)

```
\[
\begin{array}{l}
\text{pattern level} \\
\begin{array}{l}
\text{verb: } V_{\_\alpha} \\
\text{sem: } \alpha \\
\text{prn: } K \\
\end{array} \\
\hline
\alpha \in \{\text{that, when, where, how, why}\}. \\
\text{content level} \\
\begin{array}{l}
\text{sur: } v_2 \\
\text{cat: } \text{sub} \\
\text{sem: } \text{that} \\
\text{arg: } (\text{say 27}) \\
\text{nc: } 28 \\
\text{pc: } \text{prn: } \\
\end{array} \\
\hline
\begin{array}{l}
\text{sur: } \text{Bill} \\
\text{cat: } \text{noun: } (\text{person } y) \\
\text{sem: } \text{nm m} \\
\text{fnc: } (\text{say 27}) \\
\text{nc: } 28 \\
\text{pc: } \text{prn: } \\
\end{array}
\]
\]
```

The arrival of the subclause verb triggers the application of SBJ×PRD:

5.5.9 CROSS-COPYING bill AND believe WITH SBJ×PRD (line 6a)

```
\[
\begin{array}{l}
\text{pattern level} \\
\begin{array}{l}
\text{noun: } \alpha \\
\text{cat: } \text{NP} \\
\text{fnc: } \text{PRD} \\
\text{prn: } K \\
\end{array} \\
\hline
\begin{array}{l}
\text{verb: } \beta \\
\text{cat: } \text{NP}^{\prime} \text{ X v} \\
\text{arg: } \text{PRD} \\
\text{prn: } K \\
\end{array} \\
\hline
\begin{array}{l}
\text{noun: } \alpha \\
\text{cat: } \#\text{NP}^{\prime} \text{ X v} \\
\text{arg: } \alpha \\
\text{prn: } K \\
\end{array}
\]
\]
```

Agreement conditions as in \[2.1.2\].

```
\[
\begin{array}{l}
\text{content level} \\
\begin{array}{l}
\text{sur: } \text{believes} \\
\text{cat: } \text{sub} \\
\text{sem: } \text{believe} \\
\text{fnc: } \text{pres} \\
\text{nc: } 28 \\
\text{pc: } \text{prn: } \\
\end{array} \\
\hline
\begin{array}{l}
\text{sur: } \text{Bill} \\
\text{cat: } \text{noun: } (\text{person } y) \\
\text{sem: } \text{nm m} \\
\text{fnc: } \text{PRD} \\
\text{nc: } 28 \\
\text{pc: } \text{prn: } \\
\end{array}
\]
\]
```

With the agreement between the subject and the predicate of the subclause secured, the predicate is absorbed into the subordinating conjunction that\[14\].

\[14\] The hear mode operation sequence of SBJ×PRD (providing cross-copying with agreement) and subclause~PRD (absorbing the completed predicate into the subordinating conjunction) has been used previously in \[2.5.4, 2.5.5, 2.6.6, 2.6.7\] and \[3.5.4, 3.5.5\]. It will be used again in \[5.5.9, 5.5.10\] and in modified form in \[5.5.12, 5.5.15\].
5.5.10 ABSORBING believe INTO that with subclause □ PRD (line 6b)

\[
\begin{array}{|c|c|c|}
\hline
\text{pattern level} & \text{content level} & \text{PRD (19)} \\
\hline
\text{verb: V_n} & \text{verb: V_n} & \text{verb: V_n} \\
\text{cat: \#X' N' v} & \text{cat: \#N' X v} & \text{cat: \#N' X v} \\
\text{sem: \#that} & \text{sem: \#N' v} & \text{sem: \#that} \\
\text{arg: Y} & \text{arg: Y} & \text{arg: (person y)} \\
\text{prn: K} & \text{prn: K} & \text{prn: K} \\
\hline
\end{array}
\]

This absorption compensates the suspension of 5.5.8.

Having processed Whom does John say that Bill believes, the derivation could be completed by adding a question mark (as in line B of 5.5.3). However, because automatic word form recognition provides another that proplet, a third object clause is entered. This raises the question of whether the prn value should be (i) incremented (corresponding to line D in 5.5.3) or (ii) set to the prn value of Whom (&-marked in 5.5.3) corresponding to line C in 5.5.3.

The latter option realizes the content in 5.5.1 by stopping the unbounded object clause repetition. The final object clause is entered with PRD □ sub that

5.5.11 CROSS-COPYING believe AND that with PRD □ sub that □ (line 7)

\[
\begin{array}{|c|c|c|}
\hline
\text{pattern level} & \text{content level} & \text{PRD (19)} \\
\hline
\text{verb: \#X' N' v} & \text{verb: v_3} & \text{verb: V_n} \\
\text{cat: \#that} & \text{cat: \#N' v} & \text{cat: \#N' X v} \\
\text{sem: \#that} & \text{sem: \#that} & \text{sem: \#that} \\
\text{arg: (person y)} & \text{arg: (person y)} & \text{arg: (person y)} \\
\text{fnc: (say 27)} & \text{fnc: (say 27)} & \text{fnc: (say 27)} \\
\text{nc: \#N} & \text{nc: \#N} & \text{nc: \#N} \\
\text{pc: \#N} & \text{pc: \#N} & \text{pc: \#N} \\
\text{prn: 28} & \text{prn: 28} & \text{prn: 28} \\
\hline
\end{array}
\]

The prn variable L& picks the prn value 26& of whom at the now front, thus providing the correct prn value for the remainder of the final subclause, regardless of how many object clauses have intervened.
5.5.12 SUSPENDING v_3 AND Mary with sub\textsuperscript{clause}~SBJ (line 8)

At this point, (i) the above suspension must be compensated by combining Mary and the next word love with SBJ×PRD (5.5.13), (ii) the initial suspension of sub\textsuperscript{clause} in 5.5.5 must be compensated by combining love and whom with new WH×FV (5.5.14), and (iii) the grammatically completed love proplet must be absorbed into that with sub\textsuperscript{clause}∪PRD (5.5.15)\textsuperscript{15}

5.5.13 CROSS-COPYING Mary AND loves WITH SBJ×PRD (line 9a)

Having entered the final clausal object with PRD×sub\textsuperscript{that} in 5.5.11, the clause of the kind of verb does not matter any more, except that it must be transitive:

\textsuperscript{15} This example shows that the application of several suspension-compensating operations in the same derivation step, e.g. 9a, 9b, and 9c in 5.5.2, is not in some random order.

\textsuperscript{16} It may seem peculiar that love is not in the class of mental state verbs. The grammatical wellformedness of using love as a predicate taking a clausal object, as in ?Mary loved that John brought her flowers, is borderline.
5.5.14 CROSS-COPYING wh_1 and love with WH×FV (line 9b)

Next, the grammatically completed love is absorbed into the subordinating conjunction added in 5.5.11 (analogous to the previous application of subclause PRD in 5.5.10):

5.5.15 ABSORBING love into v_3 with subclause PRD (line 9c)

The derivation concludes with SUBIP. The unique top predicate is say. It is the only predicate without an additional fnc feature, introduced lexically into the other predicates believe and love by the subordinating conjunction that.

5.5.16 ABSORBING ? into say with SUBIP (line 10)
5. Unbounded Repetition

Agreement conditions as in 2.1.3.

Lexical lookup of the next word surface ? produces a proplet with the cat value interrog specifying the sentential mood.

We continue with point 4 of 1.5.1, i.e. the graph structure of the speak mode, based on the content which resulted from the hear mode derivation 5.5.2.

5.5.17 Graph analysis underlying production of 5.5.1

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

Grammarically, the cycle consisting of the arcs 4, 5, 11 may be repeated, whereby each repetition has new content and new arc numbers. The cycles represented here by the arcs 9-10 and 7-8, in contrast, occur only once, the first as the end of the beginning, the second as the beginning of the unbounded repetition.

The next item on the to-do list of 1.5.1 is point 5, i.e. the list of speak mode operation names with surface realizations, to be followed by the complete sequence of explicit speak mode operation applications (point 6). Point 5 shows that the graph structure of the NAG enables the derivation of the 11 word form surfaces with only five empty traversals. The difference between the number of word form surfaces and empty traversals stays the same as the number of minimal repetitions increases: 11-5=6, 14-8=6, 17-11=6, 20-14=6,...
5.5.18 SEQUENCE OF OPERATION NAMES AND SURFACE REALIZATIONS

arc 3: \(V\backslash V\) from \(say\) to \(believe\) \[5.5.19\]
arc 6: \(V\backslash V^f\) from \(believe\) to \(love\) \[5.5.20\]
arc 9: \(V\backslash WH\) from \(love\) to \(wh_1\) \[5.5.21\]
arc 10: \(WH\backslash V\) from \(wh_1\) to \(love\) \[5.5.22\]
arc 11: \(V\backslash V\) from \(love\) to \(believe\) \[5.5.23\]
arc 12: \(V\backslash V\) from \(believe\) to \(say\) \[5.5.24\]
arc 1: \(V\backslash N\) from \(say\) to \(john\) \[5.5.25\]
arc 2: \(N\backslash V\) from \(john\) to \(say\) \[5.5.26\]
arc 3: \(V\backslash V\) from \(say\) to \(believe\) \[5.5.27\]
arc 4: \(N\backslash V\) from \((person\ y)\) to \(believe\) \[5.5.28\]
arc 5: \(V\backslash V^f\) from \(believe\) to \(love\) \[5.5.29\]
arc 6: \(V\backslash N\) from \(love\) to \((person\ z)\) \[5.5.30\]
arc 7: \(N\backslash V\) from \((person\ z)\) to \(love\) \[5.5.31\]
arc 8: \(V\backslash V\) from \(love\) to \(believe\) \[5.5.32\]
arc 11: \(V\backslash V\) from \(believe\) to \(say\) \[5.5.33\]
arc 12: \(V\backslash V\) from \(love\) to \(believe\) \[5.5.34\]

To realize the first word \(Whom\), the navigation must travel from the top verb \(say\) all the way down to the last object clause—which for technical reasons has the lowest \(prn\) value, here 26, marked with & in the hear mode (5.5.4). This is based on the operations \(V\backslash V\) moving from \(say\) to \(believe\) via arc 3, \(V\backslash V^f\) from \(believe\) to \(love\) via arc 6, and \(V\backslash WH\) from \(love\) to \(wh_1\) via arc 9:

5.5.19 NAVIGATING WITH \(V\backslash V\) FROM \(say\) TO \(believe\) (arc 3)

In an unbounded repetition with an initial \(wh\)-interrogative and \(n\) object clauses, there are \(n-2\) \(V\backslash V\) applications like [5.5.19] followed by \(V\backslash V^f\) (f stand-
ing for final), followed by $V \backslash WH$. Because the present example has only three object clauses (i.e. the minimum for showing a repetition), $V \backslash V$ applies only once, i.e. it does not apply again, and the final object clause is entered directly with $V \backslash V^f$:

5.5.20 Navigating with $V \backslash V^f$ from believe to love (arc 6)

\[
V \backslash V^f \quad (38)
\]

\[
\begin{array}{l}
\text{pattern level} \\
\text{verb: } \alpha \\
\text{arg: } X \ (\beta \ L&)
\end{array}
\Rightarrow
\begin{array}{l}
\text{sur: lexverb(\(\hat{\gamma}\))} \\
\text{verb: } \beta \\
\text{fnc: } (\alpha \ K)
\end{array}
\]

The second arg value wh_1 of the goal proplet love serves as the continuation value for the new speak mode operation $V \backslash WH$ (which performs a similar function as $V \backslash N$ in 2.3.10).

5.5.21 Navigating with $V \backslash WH$ from love to wh_1 (arc 9)

\[
V \backslash WH \quad (35)
\]

\[
\begin{array}{l}
\text{pattern level} \\
\text{verb: } \alpha \\
\text{sem: } that \ X \\
\text{arg: } X \ WH \_n \\
\text{prn: } L&
\end{array}
\Rightarrow
\begin{array}{l}
\text{sur: lexnoun(\(\hat{\gamma}\))} \\
\text{noun: WH \_n} \\
\text{fnc: } \alpha \\
\text{prn: } L&
\end{array}
\]

Lexnoun uses the & marker of the prn value, the core value wh_1, and the cat value obq to realize the surface whom.

The next word to be realized is does. For this the navigation has to return empty to the top verb say to retrieve the initial sem value do. The first step on the route back is from wh_1 to love with the new operation $WH \backslash V$ via arc 10:
5.5.22 **Navigating with WH\V from wh_1 back to love** (arc 10)

\[WH\V (16)\]

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
</table>
| noun: WH_n | [verb: α

sem: that X
arg: \{ X #WH_n\}
prn: L&] |
| fnc: α | ↑ |
| prn: L& | ↓ |
| sur: | [sur: | noun: wh_1

cat: obq
sem: | fnc: love
mdr: | arg: \{ person z \#wh_1\}

fnc: (believe 28)
mdr: |
| pc: | nc: |
| prn: 26& | prn: 26& |

The next step on the empty return to the top verb *say* is from *love* to *believe* via arc 11, based on \[V\V (2.6.16)\].

5.5.23 **Navigating with V\V from love back to believe** (arc 11)

\[V\V (18)\]

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
</table>
| verb: α

sem: that Z
fnc: \(\beta \ ? \ K\)
prn: L | [sur: lexverb(\(\hat{\beta}\))
verb: \(\beta\)
arg: Y (α L)
prn: K] |
| ↑ | ↓ |
| sur: | [sur: | verb: believe

cat: \#ns3' \#a' v
sem: that \(\beta\)
arg: \{ person y \} (love 26&)

fnc: (say 27)
mdr: |
| pc: | nc: |
| prn: 26& | prn: 28 |

The use of the different prn variables L and K allows \[V\V\] to apply no matter whether the prn values are adjacent and ordered (as would be required by the variables K and K+1) or not. This allows \[V\V\] to apply once more on the way back to the top verb, using the continuation feature \[fnc: (say 27)\] of believe.

In the following traversal of arc 12 with another application of \[V\V\], the lexicalization rule lexverb in the sur slot of the goal pattern realizes the surface *does* from the cat value \#ns3' and the sem values do pres of the goal proplet *say*. 
5.5.24 Navigating with $V/V$ from `believe` back to `say` (arc 12)

The dotted underline (1.6.5, 3) of $\text{func: (} \beta \text{)} K$] allows matching constants with (e.g. 2.6.16, 5.5.33, 5.5.34) or without (e.g. 5.5.23, 5.5.24) a #-marking.

Using the operations for clausal object constructions (Sect. 2.6), the next navigation steps follow arcs 1–8 through the NAG (5.5.17), realizing remaining John say that Bill believes that Mary loves. The first step is an application of standard $V/N$:

5.5.25 Navigating with $V/N$ from `say` to `john` (arc 1)

Using the marker in the sur slot of a name, here john, lexnoun realizes the surface John, temporarily covering the name marker until the name surface is moved to the agent’s action component for word form realization, again revealing the name marker.

Next, the navigation returns to the top verb `say` using the operation $N/V$. 
5.5.26 **Navigating with $N/V$ from John back to say** (arc 2)

$$N/V \overset{\{2\}}{\Rightarrow}$$

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: $\beta$</td>
<td>sur: john</td>
</tr>
<tr>
<td>fnr: $\alpha$</td>
<td>noun: (person x)</td>
</tr>
<tr>
<td>mdr: Z</td>
<td>cat: snp</td>
</tr>
<tr>
<td>prn: K</td>
<td>fnc: say</td>
</tr>
<tr>
<td>Z is NIL, or</td>
<td>mdr:</td>
</tr>
<tr>
<td>elementary and</td>
<td>nc:</td>
</tr>
<tr>
<td>#-marked</td>
<td>pc:</td>
</tr>
<tr>
<td></td>
<td>prn: 27</td>
</tr>
</tbody>
</table>

Lexverb uses the core value *say* and the *sem* value *do* for realizing the surface in the goal proplet. The result of the instruction shows up in the goal proplet of **5.5.27**.

The traversal of arc 3 accesses the verb of the first object sentence with $V\backslash V$:

5.5.27 **Navigating with $V\backslash V$ from say to believe** (arc 3)

$$V\backslash V \overset{\{7\}}{\Rightarrow}$$

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb: $\alpha$</td>
<td>sur: that</td>
</tr>
<tr>
<td>arg: X ($\beta K+1$)</td>
<td>verb: believe</td>
</tr>
<tr>
<td>prn: K</td>
<td>cat: #ns3' #a' v</td>
</tr>
<tr>
<td></td>
<td>sem: that pres</td>
</tr>
<tr>
<td></td>
<td>mdr:</td>
</tr>
<tr>
<td></td>
<td>nc:</td>
</tr>
<tr>
<td></td>
<td>pc:</td>
</tr>
<tr>
<td></td>
<td>prn: 27</td>
</tr>
</tbody>
</table>

Lexverb uses the initial *sem* value of the goal proplet *believe* to realize the subordinating conjunction *that* (function word precipitation; see **5.5.10** for the corresponding function word absorption in the hear mode).

The next navigation step resembles the usual beginning of a declarative sentence: by going from the verb to the first argument, $V/\backslash N$ realizes the subject, here Bill.
5. Unbounded Repetition

5.5.28 **Navigating with \( V/N \) from *believe* to *bill* (arc 4)**

\[ V/N (4) \]

\[
\begin{align*}
\text{pattern level} & : \quad \verb: \alpha \\
\text{arg} & : \beta X \\
\text{prn} & : K \\
\Rightarrow & : \quad \text{sur: lexnoun}(\beta) \\
\text{noun} & : \beta \\
\text{fnc} & : \alpha \\
\text{prn} & : K
\end{align*}
\]

\#-mark \( \beta \) in the arg slot of proplet \( \alpha \)

\[
\begin{align*}
\uparrow & : \quad \text{sur: believe} \\
\text{cat} & : \#n3' \#a' v \\
\text{sem: that} & : \text{pres} \\
\text{arg} & : (\text{person y}) \ (\text{love } 26&) \\
\text{fnc} & : \#(\text{say } 27) \\
\text{mdr} & : \text{nc} \\
\text{pc} & : \text{prn: } 28
\end{align*}
\]

\[
\begin{align*}
\downarrow & : \quad \text{sur: Bill} \\
\text{noun} & : (\text{person y}) \\
\text{cat} & : \text{snp} \\
\text{sem: nm m} \\
\text{fnc} & : \text{believe} \\
\text{mdr} & : \text{nc} \\
\text{pc} & : \text{prn: } 28
\end{align*}
\]

The result of the instruction shows up in the goal proplet of 5.5.29.

The object sentence continues like a declarative sentence: by returning from the subject to the predicate, \( N/V \) realizes the finite verb.

5.5.29 **Navigating with \( N/V \) from *bill* back to *believe* (arc 5)**

\[ N/V (5) \]

\[
\begin{align*}
\text{pattern level} & : \quad \text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{mdr: } Z \\
\text{prn: } K \\
\Rightarrow & : \quad \text{sur: lexverb}(\alpha) \\
\text{verb: } \alpha \\
\text{arg: } \#\beta X \\
\text{prn: } K
\end{align*}
\]

\#-mark \( \alpha \) in the fnc slot of proplet \( \beta \)

\[
\begin{align*}
\uparrow & : \quad \text{sur: bill} \\
\text{noun} & : (\text{person y}) \\
\text{cat} & : \text{snp} \\
\text{sem: nm m} \\
\text{fnc: } \text{believe} \\
\text{mdr} & : \text{nc} \\
\text{pc} & : \text{prn: } 28
\end{align*}
\]

\[
\begin{align*}
\downarrow & : \quad \text{sur: believes} \\
\text{verb} & : \text{believe} \\
\text{cat} & : \#n3' \#a' v \\
\text{sem: that} & : \text{pres} \\
\text{arg} & : (\text{person y}) \ (\text{love } 26&) \\
\text{fnc} & : \#(\text{say } 27) \\
\text{mdr} & : \text{nc} \\
\text{pc} & : \text{prn: } 28
\end{align*}
\]

The second arg slot of the verb proplet *believe* provides the value for continuing to the grammatical object, which happens to be another object clause, represented by the address of its predicate, i.e. (love 26&). The &-marking of its prn value, provided by the hear mode derivation (5.5.4, 5.5.11), marks love as representing (i) the final clause of the repeating object clause construction and (ii) as the functor taking the initial whom as its grammatical object. The result of the instruction shows up in the goal proplet of 5.5.30.

The first step of entering the final subclause (re)uses the operation \( V/V \) (5.5.20);
5.5.30 Navigating with \( V \backslash V f \) from \textit{believe} to \textit{love} (arc 6)

\[
\begin{array}{c}
\text{pattern level} \\
\begin{array}{l}
\text{verb: } \alpha \\
\text{arg: } X (\beta \text{ L&}) \\
\text{prn: } K
\end{array} \\
\Rightarrow \\
\begin{array}{l}
\text{sur: } \text{lexverb}(\gamma) \\
\text{verb: } \beta \\
\text{fnc: } (\alpha \text{ K}) \\
\text{prn: } \text{L&}
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\text{content level} \\
\begin{array}{l}
\text{sur: } \text{believe} \\
\text{cat: } \text{#ns3' #a' v} \\
\text{sem: } \text{that} \text{ pres} \\
\text{arg: } (\text{person y}) (\text{love } 26&) \\
\text{fnc: } (\text{say } 27) \\
\text{nc:} \\
\text{pc:} \\
\text{prn: } 28
\end{array}
\end{array}
\]

The second arg value \textit{wh_1} in the goal proplet was \#-marked by the instruction of \( V \backslash WH \) in 5.5.21.

At this point, the navigation through the content 5.5.1 has realized \textit{Whom does John say that Bill believes that}. The following two navigation steps resemble the application of \( V / N \) in 5.5.28 and of \( N / V \) in 5.5.29 in that they are like the beginning of a declarative sentence. The fact that they apply in an object clause, and moreover in the final clause of an object clause repetition, is not noticed by these operations:

5.5.31 Navigating with \( V / N \) from \textit{love} to (person z) (arc 7)

\[
\begin{array}{c}
\text{pattern level} \\
\begin{array}{l}
\text{verb: } \alpha \\
\text{arg: } \beta \text{ X} \\
\text{prn: } K
\end{array} \\
\Rightarrow \\
\begin{array}{l}
\text{sur: } \text{lexnoun}(\beta) \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K
\end{array}
\]

#-mark \( \beta \) in the arg slot of proplet \( \alpha \)

\[
\begin{array}{c}
\text{content level} \\
\begin{array}{l}
\text{sur: } \text{love} \\
\text{cat: } \text{#ns3' #a' v} \\
\text{sem: } \text{that} \text{ pres} \\
\text{arg: } (\text{person z}) \text{ #wh_1} \\
\text{fnc: } (\text{believe } 28) \\
\text{nc:} \\
\text{pc:} \\
\text{prn: } 26&
\end{array}
\end{array}
\]

Using the marker in the sur slot of a name proplet, here \textit{mary}, lexnoun realizes the surface \textit{Mary}, temporarily covering the name marker until the name surface is passed to the agent’s action component for word form realization. The result of the instruction shows up in the arg slot of the goal proplet of 5.5.32.
5.5.32 Navigating with N/V from *mary* back to *love* (arc 8)

With the second arg value *wh_1* of *love* already #-marked (5.5.21) there is no other option than taking the short route via arcs 11 and 12 back to the top verb to realize the question mark and end up in a position suitable for a possible navigation to a next proposition (1.4.5). These steps are performed by two consecutive applications of V\V:

5.5.33 Navigating with V\V from *love* to *believe* (arc 11)

Similar to [5.5.23] and [5.5.24] the different prn variables L and K match the values 26& and 28 in the current as well as the values 28 and 27 in the following application. In all four V\V applications, the relation coded by the fnC value of the input and the verb value of the output is sufficient for proper matching, i.e. it doesn’t rely or insist on consecutive prn values à la K and K+1.
5.5.34 **Navigating with \( V \setminus V \) from *believe* to *say* (arc 12)**

\[
\begin{array}{c}
\text{pattern level} \\
\uparrow \\
\text{content level} \\
\end{array}
\]

\[
\begin{array}{c}
\text{verb: } \alpha \\
\text{sem: that } Z \\
\text{inc: } (\beta, K) \\
\text{prn: } L \\
\Rightarrow \\
\text{verb: } \beta \\
\text{arg: } Y (\alpha L) \\
\text{prn: } K \\
\end{array}
\]

\[
\begin{array}{c}
\text{sur: } \text{lexverb}(\beta) \\
\text{verb: } \beta \\
\text{arg: } Y (\alpha L) \\
\text{prn: } K \\
\end{array}
\]

Lexverb uses the \textit{cat} value \textit{interrog} of *say* to realize the question mark.

Following standard procedure, we conclude with point 7 on the to-do list. Extending DBS.22 to an example class with repeating object clauses required the three hear mode operations \( WH \sim AUX \) (h13), \( do \times NOM \) (h23), and \( PRD \times \text{sub} \) (h8), and the three speak mode operations \( V \setminus WH \) (s35), \( WH \setminus V \) (s36), and \( V \setminus V \) (s38). The hear mode derivation used 13 operation applications and the speak mode used 16.

5.5.35 **Comparing hear and speak mode surface segmentation**

**hear mode:**

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6a & 6b \\
\text{Whom} & \sim & \text{does} & \times & \text{John} & \cup & \text{say} & \times & \text{that} & \sim & \text{Bill} & \times & \cup \\
7 & 8 & 9a & 9b & 9c & 10 \\
\text{believes} & \times & \text{that} & \sim & \text{Mary} & \times & \times & \cup & \text{loves} & \cup
\end{array}
\]

**speak mode:**

\[
\begin{array}{cccccccc}
3 & 6 & 9 & 10 & 11 & 12 & 1 \\
V \setminus V & V \setminus V & V \setminus WH & \text{Whom} & WH \setminus V & V \setminus V & \text{does} & V \setminus N & \text{John} \\
2 & 3 & 4 & 5 & 6 & 7 \\
N \setminus V & \text{say} & V \setminus V & \text{that} & V \setminus N & \text{Bill} & N \setminus V & \text{believes} & V \setminus V & \text{that} & V \setminus N & \text{Mary} \\
8 & 11 & 12 \\
N \setminus V & \text{loves} & V \setminus V & V \setminus V & ?
\end{array}
\]

Unlike the gapping constructions, the arc numbering of the NAG is not breadth first, but depth first. There are four multiple traversals, namely 3, 6, 11, and 12, resulting in the total of 16 speak mode operation applications.
5.6 DBS.24: Repeating Adnominal Clauses

Like repeating object clauses, repeating adnominal clauses are extrapositional and each is embedded into a next-higher clause. Consider an example with two adnominal clauses, each with a subject gap, and its content:

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

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{sur: saw} & \text{noun: (person x)} & \text{cat: snp} & \text{sem: nm f} & \text{fnc: see} & \text{prn: 30} \\
\hline
\text{sur: who} & \text{verb: love} & \text{cat: #n3' #a v} & \text{sem: def sg} & \text{fnc: who} & \text{arg: (man 29)} \\
\hline
\text{sur: Fido} & \text{noun: (dog x)} & \text{cat: snp} & \text{sem: nm} & \text{fnc: feed} & \text{prn: 31} \\
\hline
\end{array}
\]

Similar to the examples with a single adnominal clause in Sects. 3.3 (subject gap) and 3.4 (object gap), the extrapositional \text{mdr}[\text{mdd: (man 29)}] relation between \text{man} and (who) loves... is coded by the features [\text{mdr: (love 30)}] in the \text{man} proplet and [\text{mdd: (man 29)}] in the love proplet, and similarly between the \text{woman} and the feed proplet. The two occurrences of the subordinating conjunction who (aka relative pronoun) have absorbed the predicate of their subclause and show a gap, represented as \emptyset, in their first \text{arg} slot (subject). The implicit filler of the gap is shown in the \text{mdd} slot directly underneath.

Following standard procedure, we continue with point 2 on the to-do list i.e. the time-linear surface compositional hear mode derivation:

5.6.2 Graphical Hear Mode Derivation of the Content 5.6.1

\[
\begin{array}{|c|c|c|c|c|}
\hline
\text{sur: Mary} & \text{noun: (person x)} & \text{cat: snp} & \text{sem: nm f} & \text{arg: (person x)} \\
\hline
\text{sur: the} & \text{noun: man} & \text{cat: #n' #a decl} & \text{sem: past} & \text{fnc: who} \\
\hline
\text{sur: loves} & \text{verb: love} & \text{cat: #n3' #a v} & \text{sem: def sg} & \text{arg: (man 29)} \\
\hline
\text{sur: who} & \text{verb: fed} & \text{cat: #n' #a v} & \text{sem: past} & \text{fnc: feed} \\
\hline
\text{sur: Fido} & \text{noun: (dog x)} & \text{cat: snp} & \text{sem: nm} & \text{fnc: feed} \\
\hline
\end{array}
\]

17 As a grammatical phenomenon, embedding one clause into another does not require the formal method of substitution, as in recursion. In DBS, clausal embedding is treated like any other semantic relation of structure between two proplets: it is realized by semantically meaningful core and continuation attributes and by values defined as address. This holds not only for the single embeddings of clausal subjects (Sect. 5.5), clausal objects (Sect. 5.6), clausal adnominals (Sects. 3.3, 3.4), and clausal adverbials (Sect. 3.5), but also for the repeated clausal embeddings in Sects. 5.5 and 5.6.
The initial sentence Mary saw the man is completed in line 3. In line 4, man is extended with the beginning of a clausal adnominal by adding who. In line 5, the two occurrences of \( v_1 \) (i.e. the core value of who) are replaced by the core value of love (function word absorption with simultaneous substitution). Lines 6 and 7 complete the clausal adnominal, resulting in Mary saw the man who loves the woman. In line 8, the addition of a second who extends woman with the beginning of another clausal adnominal with a subject gap, namely who fed Fido. The repeated addition of adnominal subclauses has no grammatical limit, but our example of an adnominal clause repetition is minimal in that there is only one.

Following standard procedure, we continue with point 3 on the DBS to-do list \( \{5,1\} \) i.e. the complete sequence of explicit hear mode operation applications. The first two input surfaces Mary and saw are analyzed as proplets by automatic word form recognition and stored at the now front. The second proplet, i.e. see, activates an operation, here SBJ×PRD, matching its second input pattern. The activated operation looks at the now front for a proplet matching its first input pattern, here mary, and applies:

### 5.6.3 Cross-copying Mary and saw with SBJ×PRD (line 1)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: ( \alpha )</td>
<td>sur: Mary</td>
</tr>
<tr>
<td>cat: NP</td>
<td>noun: (person x)</td>
</tr>
<tr>
<td>fn:</td>
<td>cat: snp</td>
</tr>
<tr>
<td>sur: verb see</td>
<td>sem: mm f</td>
</tr>
<tr>
<td>arg: (p.x)man</td>
<td>mdr:</td>
</tr>
<tr>
<td>prn: K</td>
<td>pc:</td>
</tr>
<tr>
<td>fnc: see</td>
<td>prn: 29</td>
</tr>
<tr>
<td>prn:</td>
<td></td>
</tr>
</tbody>
</table>

| Agreement conditions as in [\( . \) ] |

Next, automatic word recognition analyzes the third surface the as a determiner proplet and stores it at the now front. The current next word matches the second input pattern of PRD×OBJ. The activated operation looks at the now
for a proplet matching its first input pattern. It finds see and applies as follows:

### 5.6.4 Cross-copying saw and the with PRD × OBJ (line 2)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
<th>PRD × OBJ (line 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[verb: ( \beta ) ( \text{cat: } #X N' Y v ) ( \text{fnc: } ) ( \text{prn: } K )]</td>
<td>[sur: see ( \text{cat: } #n' a' v ) ( \text{sem: } ) ( \text{arg: (person x) } ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: 29} )]</td>
<td>[noun: ( \alpha ) ( \text{cat: } #CN' N ) ( \text{fnc: } ) ( \text{prn: K} ) ]</td>
</tr>
<tr>
<td>( \Rightarrow )</td>
<td>( \Uparrow )</td>
<td>( \Rightarrow )</td>
</tr>
<tr>
<td>[verb: ( \beta ) ( \text{cat: } #X' #N' Y v ) ( \text{fnc: } ) ( \text{prn: } K ) ]</td>
<td>[sur: the ( \text{cat: } #n_1 ) ( \text{sem: } ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: } ) ]</td>
<td>[sur: see ( \text{cat: } #n' #a' v ) ( \text{sem: } ) ( \text{arg: (person x) } ) ( \text{n}_1 ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: 29} ) ]</td>
</tr>
<tr>
<td>[noun: ( \alpha ) ]</td>
<td>[noun: ( n_1 ) ]</td>
<td>[noun: ( n_1 ) ]</td>
</tr>
<tr>
<td>[cat: X CN’ NP ( \text{sem: } Y ) ( \text{prn: } K ) ]</td>
<td>[cat: SN ( \text{sem: } ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: } ) ]</td>
<td>[cat: X NP ( \text{sem: } Y Z ) ( \text{prn: } K ) ]</td>
</tr>
<tr>
<td>[Agreement conditions as in 4.5.4]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The fourth surface man provided by the external input is analyzed by automatic word form recognition. It matches the second input pattern of DET∪CN:

### 5.6.5 Absorbing man into the with DET∪CN (line 3)

<table>
<thead>
<tr>
<th>Pattern level</th>
<th>Content level</th>
<th>DET∪CN (line 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[noun: N ( n ) ( \text{cat: } X N' ) ( \text{sem: } Y ) ( \text{prn: } K ) ]</td>
<td>[sur: ( n_2 ) ( \text{cat: } #n'_n ) ( \text{sem: } ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: 29} ) ]</td>
<td>[noun: ( \alpha ) ( \text{cat: } X ) ( \text{sem: } Z ) ( \text{prn: } K ) ]</td>
</tr>
<tr>
<td>( \Uparrow )</td>
<td>( \Uparrow )</td>
<td>( \Downarrow )</td>
</tr>
<tr>
<td>[noun: ( \alpha ) ]</td>
<td>[sur: man ( \text{cat: } #SN ) ( \text{sem: } ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: } ) ]</td>
<td>[sur: ( \text{cat: } #SN' #a' ) ( \text{sem: } ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: 29} ) ]</td>
</tr>
<tr>
<td>[cat: X NP ( \text{sem: } Y ) ( \text{prn: } K ) ]</td>
<td>[cat: SN ( \text{sem: } ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: } ) ]</td>
<td>[cat: SN' ( \text{sem: } ) ( \text{fnc: } ) ( \text{pc: } ) ( \text{prn: 29} ) ]</td>
</tr>
<tr>
<td>[Agreement conditions as in 2.2.4]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Once DET∪CN has fused the man and the determiner proplet into one (function word absorption), the two input proplets are replaced at the now front by the output proplet.

At this point, the hear mode derivation has interpreted the proposition Mary saw the man with the prn value 29, consisting of three order-free proplets. Next automatic word form recognition provides the proplet \( v_1 \) for who, which matches the second input pattern of the operation CN×sub\( \text{who} \) (3.3.4):
5.6.6 CROSS-COPYING man and who with CN × sub\(^{who}\) (line 4)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \beta \epsilon \{\text{who, which}\}. \text{Agreement conditions as in 3.3.4}\end{array}
\]

\[
\begin{array}{c|c|c}
\text{CN} \times \text{sub}^{who} \quad & \qquad & \qquad \\
\begin{array}{c}
\text{noun: } \alpha \\
\text{cat: NP} \\
\text{mdr: (man 29)} \\
\text{prn: K}
\end{array} & \rightarrow & \begin{array}{c}
\text{noun: } \alpha \\
\text{cat: NP'} \\
\text{mdr: (V_n K+1)} \\
\text{prn: K}
\end{array}
\end{array}
\]

\[
\begin{array}{c|c|c}
\begin{array}{c}
\text{noun: man} \\
\text{cat: snp} \\
\text{sem: def sg} \\
\text{fnc: see} \\
\text{mdr: (man 29)} \\
\text{pc: } \text{prn: 29}
\end{array} & \begin{array}{c}
\text{noun: man} \\
\text{cat: np} \\
\text{sem: def sg} \\
\text{fnc: see} \\
\text{mdr: (V_1 30)} \\
\text{pc: } \text{prn: 29}
\end{array} & \begin{array}{c}
\text{noun: man} \\
\text{cat: np} \\
\text{sem: def sg} \\
\text{fnc: see} \\
\text{mdr: (man 29)} \\
\text{pc: } \text{prn: 30}
\end{array}
\end{array}
\]

CN \times sub\(^{who}\) increments the prn value of the subordinating conjunction who as the beginning of the adnominal modifier clause. The lexical entry uses the English surface as the first value of the sem attribute to specify the kind of conjunction and contributes \(\emptyset\) as the first (subject) arg value. The value \(v_1\) is copied into the mdr slot of man and man is copied into the mdd slot of \(v_1\).

Next sub\(^{who}\cup PRD (3.3.5) absorbs the verb love of the first adnominal clause into the subordinating conjunction:

5.6.7 ABSORBING loves into who with sub\(^{who}\cup PRD (line 5)

\[
\begin{array}{c|c|c}
\text{pattern level} & \text{content level} & \alpha \epsilon \{\text{who, which}\}. \text{Agreement conditions as in 3.3.5}\end{array}
\]

\[
\begin{array}{c|c|c}
\text{sub}^{who}\cup \text{PRD} (3.3.5) & \qquad & \qquad \\
\begin{array}{c}
\text{verb: } \beta \\
\text{cat: NP'} \\
\text{sem: Y} \\
\text{arg: } \emptyset \\
\text{mdr: (γ K-1)} \\
\text{prn: K}
\end{array} & \rightarrow & \begin{array}{c}
\text{verb: } \beta \\
\text{cat: #NP' X v} \\
\text{sem: } \alpha Y \\
\text{arg: } \emptyset \\
\text{mdr: (γ K-1)} \\
\text{prn: K}
\end{array}
\end{array}
\]

\[
\begin{array}{c|c|c}
\begin{array}{c}
\text{sur: } \text{verb: } v_1 \\
\text{cat: ns3', n'} \\
\text{sem: who} \\
\text{arg: } \emptyset \\
\text{mdr: (man 29)} \\
\text{pc: } \text{prn: 30}
\end{array} & \begin{array}{c}
\text{sur: loves} \\
\text{verb: love} \\
\text{cat: ns3' a' v} \\
\text{sem: pres} \\
\text{arg: } \emptyset \\
\text{mdr: (man 29)} \\
\text{pc: } \text{prn: 30}
\end{array} & \begin{array}{c}
\text{sur: } \text{verb: love} \\
\text{cat: #ns3' a' v} \\
\text{sem: who pres} \\
\text{arg: } \emptyset \\
\text{mdr: (man 29)} \\
\text{pc: } \text{prn: 30}
\end{array}
\end{array}
\]

As shown in line 5 of 5.6.2 simultaneous substitution replaces \(v_1\) with love also in the mdr slot of man.
The hear mode derivation has now interpreted Mary saw the man who loves. The adnominal clause with the prn value 30 is continued by adding the grammatical object the woman to the transitive verb love.

5.6.8 Cross-copying loves and the with PRD × OBJ (line 6)

The filled valency positions of love are shown as [arg: ∅ n_2].

Next, the place-holder variable n_2 is replaced with the constant value woman by simultaneous substitution:

5.6.9 Absorbing woman into n_2 with DET ∪ CN (line 7)

With the completion of the first adnominal subclause modifying the grammatical object man in the main clause, the grammatical object woman may itself be extended with a second adnominal clause (first and only repetition), beginning with the subordinating conjunction (similar to 5.6.6):
5. Unbounded Repetition

5.6.10 CROSS-COPYING woman and who with CN × sub\textsuperscript{who} (line 8)

\textbf{CN × sub\textsuperscript{who} (line 8)}

\begin{tabular}{|c|c|}
\hline
noun: & \alpha \\
cat: & NP \\
mdd: & (γ K-1) \\
prn: & K \\
\hline
\end{tabular}

\begin{tabular}{|c|c|}
\hline
verb: & V_n \\
cat: & [NP'] \\
sem: & α \\
arg: & ∅ \\
prn: & K+1 \\
\hline
\end{tabular}

$\Rightarrow$

\begin{tabular}{|c|c|}
\hline
noun: & \alpha \\
cat: & NP \\
mdd: & (V_n K+1) \\
prn: & K \\
\hline
\end{tabular}

$\beta \epsilon \{\text{who, which}\}$. Agreement conditions as in \[3.3.4\]

At this point, the hear mode derivation has interpreted Mary saw the man who loves the woman who. Using the pattern [prn: K+1], the operation has incremented the prn value of the latest subordinating conjunction to 31.

The derivation continues intrapropositionally with sub\textsuperscript{who} \cup PRD, which absorbs the verb of the adnominal subclause (similar to \[3.3.5\] and \[5.6.7\]):

5.6.11 ABSORBING fed into who with sub\textsuperscript{who} \cup PRD (line 9)

\textbf{sub\textsuperscript{who} \cup PRD (line 9)}

\begin{tabular}{|c|c|}
\hline
verb: & V_n \\
cat: & [NP'] \\
sem: & α \\
arg: & ∅ \\
prn: & K \\
\hline
\end{tabular}

\begin{tabular}{|c|c|}
\hline
verb: & β \\
cat: & [NP' X v] \\
sem: & γ Y \\
arg: & ∅ \\
prn: & (γ K-1) \\
\hline
\end{tabular}

$\Rightarrow$

\begin{tabular}{|c|c|}
\hline
verb: & β \\
cat: & #NP' X v \\
sem: & γ Y \\
arg: & ∅ \\
prn: & (γ K-1) \\
\hline
\end{tabular}

$\alpha \epsilon \{\text{who, which}\}$. Agreement conditions as in \[3.3.5\]

\begin{tabular}{|c|c|}
\hline
sur: & verb: v_2 \\
cat: & [ns3', n'] \\
sem: & who \\
arg: & ∅ \\
mdd: & (woman 30) \\
nc: & pc: & prn: & 30 \\
\hline
\end{tabular}

\begin{tabular}{|c|c|}
\hline
sur: & verb: feed \\
cat: & #n a' v \\
sem: & who past \\
arg: & ∅ \\
mdd: & (woman 30) \\
nc: & pc: & prn: & 31 \\
\hline
\end{tabular}

There is no suspension in the hear mode derivation of an adnominal subclause with a subject gap, repeated or not (\[3.3.2\]), in contradistinction to adnominal subclauses with an object gap (\[3.4.2\] line 3).
To complete the second adnominal subclause, standard \textit{PRD} \textit{\times OBJ} applies and combines \textit{feed} and \textit{fido}:

5.6.12 \textbf{CROSS-COPYING \textit{fed} AND \textit{Fido} WITH \textit{PRD} \textit{\times OBJ} (line 10)}

\[
\begin{array}{c}
\text{\textbf{PRD} \textit{\times OBJ} (12)} \\
\begin{array}{c|c|c}
\text{pattern} & \text{content} & \text{level} \\
\hline
\text{verb: } & \text{noun: } & \Rightarrow \\
\beta & \alpha & \\
\text{cat: } & \text{cat: } & \\
#X' & CN' & \\
\text{NP' Y v} & \text{NP} & \\
\text{arg: } & \text{arg: } & \\
Z & Z & \\
\text{prn: } & \text{prn: } & \\
K & K & \\
\text{fnc: } & \text{fnc: } & \\
\beta & \beta & \\
\end{array}
\end{array}
\]

Given that the current adnominal clause is not further expanded by another adnominal clause modifying the object, we could have used an intransitive verb, as in \textit{who smiled} instead of transitive \textit{who fed Fido}.

The hear mode derivation concludes with an application of standard \textit{S\cup IP}:

5.6.13 \textbf{ABSORBING \textbullet{} INTO \textit{see} WITH \textit{S\cup IP} (line 11)}

\[
\begin{array}{c}
\text{\textbf{S\cup IP} (13)} \\
\begin{array}{c|c|c}
\text{pattern} & \text{content} & \text{level} \\
\hline
\text{verb: } & \text{verb: } & \Rightarrow \\
\beta & \beta & \\
\text{cat: } & \text{cat: } & \\
#X' & VT & \\
\text{prn: } & \text{prn: } & \\
K & K & \\
\text{arg: } & \text{arg: } & \\
#a' & #a' & \\
\text{v} & \text{v} & \\
\text{sem: } & \text{sem: } & \\
past & past & \\
\text{mdd: } & \text{arg: } & \\
\text{(woman 30)} & (dog x) & \\
\text{nc: } & \text{nc: } & \\
\text{pc: } & \text{pc: } & \\
\text{prn: } & \text{prn: } & \\
31 & 31 & \\
\end{array}
\end{array}
\]

The operation applies to the top predicate of the construction, identified by the absence of any \textit{∅} in its \textit{arg} slots (compare \textit{see} with the other verb proplets \textit{love} and \textit{feed} in 5.6.1).

Let us follow standard procedure and continue with point 4 of the to-do list [1,5,1] i.e. the canonical DBS graph analysis for the speak mode:
5.6.14 **Graph Analysis Underlying Multiple Adnominal Clauses**

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

Compared to the clausal_object\-predicate relation in 5.5.17, the present example uses a clausal_modifier\-noun relation. The love\-man relation shows love as a modifier of man, whereby the modified serves as the implicit subject of love, and similarly for the feed\-woman relation (NLC 7.3.1). The depth first numbering of the arcs (Sect. 1.4) results in the consecutive traversal numbering 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 (cf. (iv) surface realization).

Following standard procedure, we continue with point 5 of 1.5.1, i.e. the list of speak mode operations with surface realizations:

**5.6.15 Sequence of Operation Names and Surface Realizations**

<table>
<thead>
<tr>
<th>Arc</th>
<th>Operation</th>
<th>From</th>
<th>To</th>
<th>Surface Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V/N</td>
<td>see</td>
<td>Mary</td>
<td>5.6.16</td>
</tr>
<tr>
<td>2</td>
<td>N/V</td>
<td>mary</td>
<td>saw</td>
<td>5.6.17</td>
</tr>
<tr>
<td>3</td>
<td>V/N</td>
<td>man</td>
<td>the_man</td>
<td>5.6.18</td>
</tr>
<tr>
<td>4</td>
<td>N/V</td>
<td>man</td>
<td>who_loves</td>
<td>5.6.19</td>
</tr>
<tr>
<td>5</td>
<td>V/N</td>
<td>love</td>
<td>the_woman</td>
<td>5.6.20</td>
</tr>
<tr>
<td>6</td>
<td>N/V</td>
<td>woman</td>
<td>who_fed</td>
<td>5.6.21</td>
</tr>
<tr>
<td>7</td>
<td>V/N</td>
<td>woman</td>
<td>who_fed</td>
<td>5.6.22</td>
</tr>
<tr>
<td>8</td>
<td>N/V</td>
<td>fido</td>
<td>to</td>
<td>5.6.23</td>
</tr>
<tr>
<td>9</td>
<td>V/N</td>
<td>man</td>
<td>to</td>
<td>5.6.24</td>
</tr>
<tr>
<td>10</td>
<td>N/V</td>
<td>man</td>
<td>to</td>
<td>5.6.25</td>
</tr>
<tr>
<td>11</td>
<td>V/N</td>
<td>love</td>
<td>to</td>
<td>5.6.26</td>
</tr>
<tr>
<td>12</td>
<td>N/V</td>
<td>man</td>
<td>to</td>
<td>5.6.27</td>
</tr>
</tbody>
</table>

The first three traversals are based on standard applications of V/N, N/V, and V\-N, similar to the beginning of the example in Sect. 5.1.
5.6.16 Navigating with $V/N$ from *see* to *mary* (arc 1)

\[
\begin{align*}
\text{pattern level} & : \\
\verb: \alpha & \quad \Rightarrow \\
\text{arg: } \beta & \\
\text{prn: } K &
\end{align*}
\]

\[
\begin{align*}
\text{content level} & : \\
\text{sur: } & \\
\verb: see & \\
\text{cat: } & \\
\text{sem: } & \\
\text{arg: } & \\
\text{mrd: } & \\
\text{nc: } & \\
\text{pc: } & \\
\text{prn: } 29 &
\end{align*}
\]

5.6.17 Navigating with $N/V$ from *mary* back to *see* (arc 2)

\[
\begin{align*}
\text{pattern level} & : \\
\text{noun: } \beta & \\
\text{fnc: } \alpha & \quad \Rightarrow \\
\text{arg: } & \\
\text{prn: } K &
\end{align*}
\]

\[
\begin{align*}
\text{content level} & : \\
\text{sur: } & \\
\verb: see & \\
\text{cat: } & \\
\text{sem: } & \\
\text{arg: } & \\
\text{mrd: } & \\
\text{nc: } & \\
\text{pc: } & \\
\text{prn: } 29 &
\end{align*}
\]

5.6.18 Navigating with $V\backslash N$ from *see* to *man* (arc 3)

\[
\begin{align*}
\text{pattern level} & : \\
\verb: \alpha & \quad \Rightarrow \\
\text{arg: } #\beta & \\
\text{prn: } K &
\end{align*}
\]

\[
\begin{align*}
\text{content level} & : \\
\text{sur: } & \\
\verb: see & \\
\text{cat: } & \\
\text{sem: } & \\
\text{arg: } & \\
\text{mrd: } & \\
\text{nc: } & \\
\text{pc: } & \\
\text{prn: } 29 &
\end{align*}
\]
Instead of using $\mathbf{N} \downarrow \mathbf{V}$ to return upward to the predicate, as in \ref{2.3.7} (arc4), the navigation continues downward with $\mathbf{N} \downarrow \mathbf{V}$ (3.3.12) and moves from *man* to the verb of the first adnominal clause:

**5.6.19 NAVIGATING WITH $\mathbf{N} \downarrow \mathbf{V}$ FROM *man* TO *love* (arc 4)**

\[
\begin{array}{ll}
\text{pattern level} & \text{content level} \\
\text{noun: } \alpha & \text{noun: } \beta \\
\text{mdr: } (\beta K+1) & \text{mdr: } (\alpha K) \\
\text{prn: } K & \text{prn: } K+1 \\
\text{sur: } \text{lexverb}(\beta) & \text{sur: } \text{who_loves} \\
\text{verb: } \beta & \text{verb: } \text{love} \\
\text{fnc: } \text{#see} & \text{sem: } \text{who pres} \\
\text{mdr: } (\text{love 30}) & \text{arg: } \emptyset \text{ woman} \\
\text{nc: } & \text{nc: } \\
\text{pc: } & \text{pc: } \\
\text{prn: 29} & \text{prn: 30} \\
\end{array}
\]

Lexverb realizes *who_loves* based on the feature $[\text{sem: who pres}]$, the core value *love*, the cat value $\#\text{ns3}'$, and the initial arg value $\emptyset$ of the goal proplet.

The traversal of the first adnominal subclause is completed with an application of standard $\mathbf{V \downarrow N}$:

**5.6.20 NAVIGATING WITH $\mathbf{V \downarrow N}$ FROM *love* TO *woman* (arc 5)**

\[
\begin{array}{ll}
\text{pattern level} & \text{content level} \\
\text{verb: } \alpha & \text{verb: } \text{love} \\
\text{arg: } \#X \text{ Y} & \text{cat: } \#\text{ns3'} \#a' \text{ decl} \\
\text{prn: } K & \text{sem: } \text{pres} \\
\text{fnc: } \alpha & \text{arg: } \emptyset \text{ woman} \\
\text{mdr: } & \text{mdr: } (\text{feed 31}) \\
\text{nc: } & \text{nc: } \\
\text{pc: } & \text{pc: } \\
\text{prn: 30} & \text{prn: 30} \\
\end{array}
\]

Moving along the semantic relations of structure established in the hear mode, the speak mode derivation has now traversed the proplets *mary, see, man, love, woman*, and realized the surface *Mary saw the man who loves the woman*.

The derivation continues with $\mathbf{N} \downarrow \mathbf{V}$, first applied in \ref{5.6.19} and now repeated, to navigate to the proplet *feed*. Lexverb realizes the surface *who_fed* based on the sem values *who past*, and the core value of *feed*:  

\[
\begin{array}{ll}
\text{pattern level} & \text{content level} \\
\text{noun: } \beta & \text{noun: } \text{woman} \\
\text{mdr: } (\text{feed 31}) & \text{mdr: } (\text{love 30}) \\
\text{prn: } 30 & \text{prn: 30} \\
\end{array}
\]
5.6.21 Navigating with $N \downarrow V$ from *woman* to *feed* (arc 6)

The traversal of the second adnominal subclause is completed with an application of intrapropositional $V \downarrow N$, similar to 5.6.18 and 5.6.20.

5.6.22 Navigating with $V \downarrow N$ from *feed* to *fido* (arc 7)

Using the marker in the sur slot of a name proplet, here fido, lexnoun realizes the surface *Fido*, momentarily covering the name marker until the name surface has been passed to the agent’s action component for word form realization.

It remains to return to the top predicate *see* to realize the punctuation mark and to reach a location potentially suitable for navigating along a coordination relation to a next proposition (if the nc slot has been provided with an extrapropositional value in the hear mode). This return, which is empty except for the last step, is illustrated graphically in 5.6.14 and performed by standard applications of $N \downarrow V$ (arc 8), $V \uparrow N$ (arc 9), $N \downarrow V$ (arc 10), $V \uparrow N$ (arc 11), and $N \downarrow V$ (arc 12).
5.6.23 NAVIGATING WITH N\V FROM fido BACK TO feed (arc 8)

\[
\text{N}\backslash V (\text{III}) \\
pattern level \\
\begin{align*}
\text{nouns: } & \beta \\
\text{fnr: } & \alpha \\
\text{mhr: } & Z \\
\text{prn: } & K
\end{align*}
\Rightarrow \begin{align*}
\text{sur: } & \text{lexverb}(\hat{\alpha}) \\
\text{verb: } & \alpha \\
\text{arg: } & \#X \#\beta Y \\
\text{prn: } & K
\end{align*}
\]

\[
\begin{align*}
\text{Z is } & \#\text{-marked or NIL} \\
\end{align*}
\]

\[
\uparrow \quad \downarrow
\]
\[
\begin{align*}
\text{sur: } & \text{fido} \\
\text{nouns: } & \text{(dog } x) \\
\text{cat: } & \text{snp} \\
\text{sem: } & \text{def sg} \\
\text{fnr: } & \text{feed} \\
\text{mhr: } & \\
\text{nc: } & \\
\text{prn: } & 31
\end{align*}
\quad \begin{align*}
\text{sur: } & \text{verb: } \text{feed} \\
\text{cat: } & \text{#n' #a' decl} \\
\text{sem: } & \text{past} \\
\text{arg: } & \emptyset \#(\text{dog } x) \\
\text{mhr: } & \text{(woman } 30) \\
\text{nc: } & \\
\text{prn: } & 31
\end{align*}
\]

5.6.24 NAVIGATING WITH V↑N FROM feed BACK TO woman (arc 9)

\[
\text{V}\uparrow N (\text{III}) \\
pattern level \\
\begin{align*}
\text{verbs: } & \beta \\
\text{mdd: } & (\alpha K) \\
\text{prn: } & K+1
\end{align*}
\Rightarrow \begin{align*}
\text{nouns: } & \alpha \\
\text{mhr: } & (\beta K+1) \\
\text{prn: } & K
\end{align*}
\]

\[
\uparrow \quad \downarrow
\]
\[
\begin{align*}
\text{sur: } & \text{verb: } \text{feed} \\
\text{cat: } & \#n' \#a' \text{ decl} \\
\text{sem: } & \text{past} \\
\text{arg: } & \emptyset \#(\text{dog } x) \\
\text{mhr: } & \text{(woman } 30) \\
\text{nc: } & \\
\text{prn: } & 31
\end{align*}
\quad \begin{align*}
\text{sur: } & \text{nouns: } \text{woman} \\
\text{cat: } & \text{snp} \\
\text{sem: } & \text{indef sg} \\
\text{fnr: } & \text{love} \\
\text{mhr: } & \#(\text{feed } 31) \\
\text{nc: } & \\
\text{prn: } & 30
\end{align*}
\]

5.6.25 NAVIGATING WITH N\V FROM woman BACK TO love (arc 10)

\[
\text{N}\backslash V (\text{III}) \\
pattern level \\
\begin{align*}
\text{nouns: } & \beta \\
\text{fnr: } & \alpha \\
\text{mhr: } & Z \\
\text{prn: } & K
\end{align*}
\Rightarrow \begin{align*}
\text{sur: } & \text{lexverb}(\hat{\alpha}) \\
\text{verb: } & \alpha \\
\text{arg: } & \#X \#\beta Y \\
\text{prn: } & K
\end{align*}
\]

\[
\uparrow \quad \downarrow
\]
\[
\begin{align*}
\text{sur: } & \text{nouns: } \text{woman} \\
\text{cat: } & \text{snp} \\
\text{sem: } & \text{indef sg} \\
\text{fnr: } & \text{feed} \\
\text{mhr: } & \#(\text{love } 30) \\
\text{nc: } & \\
\text{prn: } & 30
\end{align*}
\quad \begin{align*}
\text{sur: } & \text{verbs: } \text{love} \\
\text{cat: } & \#ns3' \#a' \nu \\
\text{sem: } & \text{who pres} \\
\text{arg: } & \emptyset \#\text{woman} \\
\text{mdd: } & \text{(man } 29) \\
\text{nc: } & \\
\text{prn: } & 30
\end{align*}
\]
5.6.26 **Navigating with V↑N from love back to man** (arc 11)

\[
\begin{align*}
\text{pattern level} & : \quad \text{verb: } \beta \\
& \quad \text{mdd: } (\alpha K) \\
& \quad \text{prn: } K+1 \\
\uparrow & \\
\text{content level} & : \quad \text{sur: } \text{love} \\
& \quad \text{cat: } \#ns3' \#a' \text{ v} \\
& \quad \text{sem: } \text{who} \text{ pres} \\
& \quad \text{arg: } \emptyset \#\text{woman} \\
& \quad \text{mdd: } (\text{man } 29) \\
& \quad \text{nc: } \\
& \quad \text{pc: } \\
& \quad \text{prn: } 30
\end{align*}
\]

\[
\begin{align*}
\Rightarrow & \\
\text{pattern level} & : \quad \text{verb: } \alpha \\
& \quad \text{mdd: } (\beta K+1) \\
& \quad \text{prn: } K \\
\downarrow & \\
\text{content level} & : \quad \text{sur: } \text{man} \\
& \quad \text{cat: } \text{snp} \\
& \quad \text{sem: } \text{indef sg} \\
& \quad \text{fnc: } \#\text{see} \\
& \quad \text{mdr: } \#(\text{love } 30) \\
& \quad \text{nc: } \\
& \quad \text{pc: } \\
& \quad \text{prn: } 29
\end{align*}
\]

5.6.27 **Navigating with N↓V from man back to see** (arc 12)

\[
\begin{align*}
\text{pattern level} & : \quad \text{noun: } \beta \\
& \quad \text{fnc: } \alpha \\
& \quad \text{mdd: } Z \\
& \quad \text{prn: } K \\
\uparrow & \\
\text{content level} & : \quad \text{sur: } \text{man} \\
& \quad \text{cat: } \text{snp} \\
& \quad \text{sem: } \text{indef sg} \\
& \quad \text{fnc: } \#\text{see} \\
& \quad \text{mdr: } \#(\text{love } 30) \\
& \quad \text{nc: } \\
& \quad \text{pc: } \\
& \quad \text{prn: } 29
\end{align*}
\]

\[
\begin{align*}
\Rightarrow & \\
\text{pattern level} & : \quad \text{verb: } \alpha \\
& \quad \text{arg: } \#X \#\beta Y \\
& \quad \text{prn: } K \\
\downarrow & \\
\text{content level} & : \quad \text{sur: } \text{lexverb(\hat{\alpha})} \\
& \quad \text{verb: } \alpha \\
& \quad \text{Sem: } \text{past} \\
& \quad \text{arg: } \#(\text{person x}) \#\text{man} \\
& \quad \text{mdr: } \\
& \quad \text{nc: } \\
& \quad \text{pc: } \\
& \quad \text{prn: } 29
\end{align*}
\]

The goal proplet *see* is recognized by lexverb as the top predicate because its **cat** slot concludes with a mood marker and its **arg** slot contains no \(\emptyset\) value, in contradistinction to the other predicates *love* and *feed*. Using the **cat** value **decl**, lexverb to realize the period.

Following standard procedure, we conclude with point 7 on the to-do list [1.5.1]. Extending DBS.23 to an example class with repeating adnominal clauses did not require the definition of any additional operations. The hear mode derivation used 11 operation applications and the speak mode used 12.
5.6.28 COMPARING HEAR AND SPEAK MODE SURFACE SEGMENTATION

*hear mode:*

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\times & saw & \times & the & \times & man & \cup & who & \times & loves & \cup & the & \times \\
8 & 9 & 10 & 11 \\
\end{array}
\]

Mary \times saw \times the \times man \cup who \times loves \cup the \times 

woman \cup who \times fed \times Fido \cup .

*speak mode:*

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 \\
V/N & Mary & V/N & saw & V/N & the_man & N/V & who_loves & V/N & the__woman \\
6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\end{array}
\]

As shown also by the surface realization in 5.6.14, the arc numbers of the speak mode operations are consecutive. Unlike the gapping constructions, the arc numbering of the NAG is not breadth first, but depth first as in most other constructions. There are no multiple traversals.
6. The DBS Grammars

This chapter summarizes the hear and the speak mode operations which resulted from analyzing the 24 constructions in Chaps. 2–5. Sect. 6.1 reviews the three basic formats of the hear mode operations as abstract schemata. Sect. 6.2 explores how the operations should be ordered. Sect. 6.3 presents the 43 operations of DBS.Hear. Sect. 6.4 reviews the four basic formats of the speak mode operations as abstract schemata and explains the principles of their order. Sect. 6.5 presents the 39 operations of DBS.Speak. Sect. 6.6 shows the low computational complexity of the DBS.Hear and DBS.Speak grammars.

6.1 Three Operation Formats of DBS.Hear

A hear mode operation consists of two input patterns and one or two output patterns. The name of such an operation, e.g. PRD × OBJ (2.3.4), specifies its linguistic function with two input patterns, here PRD for predicate and OBJ for grammatical object, and an intervening connective which indicates how the output is constructed from the input. There are three connectives for the three general kinds of time-linear hear mode combination in DBS, namely (i) × for cross-copying, (ii) ∼ for suspension, and (iii) ∪ for absorption.

Cross-copying establishes a semantic relation between two input proplets by copying the core value of one into a continuation slot of the other:

6.1.1 Schematic Application of a Cross-Copying Operation (×)

\[
\begin{array}{c}
\text{pattern level} \\
\text{core attribute: dog} \\
\text{prn: 23} \\
\text{condition} \\
\uparrow \\

\begin{array}{c}
\text{content level} \\
\text{core attribute: bark} \\
\text{cont. attr.: bark} \\
\text{prn: 23} \\
\end{array}
\end{array}
\Rightarrow
\begin{array}{c}
\text{pattern level} \\
\text{core attribute: bark} \\
\text{prn: 23} \\
\text{condition} \\
\downarrow \\

\begin{array}{c}
\text{content level} \\
\text{core attribute: dog} \\
\text{cont. attr.: bark} \\
\text{prn: 23} \\
\end{array}
\end{array}
\]
By binding the values \textit{dog} and \textit{bark} to the variables $\alpha$ and $\beta$, respectively, in the input patterns, the output patterns copy them into the respective continuation slots, establishing the semantic relation of structure at the content level.

A suspension operation ($\sim$) resembles cross-copying insofar as it has two input and two output patterns. Also, it assigns a \texttt{prn} value to the second input proplet (next word). It differs in that it does not define a semantic relation between the input proplets; instead, establishing the semantic relation is postponed (suspended) until the required item is provided by automatic word form recognition in the time-linear input sequence (suspension compensation).

### 6.1.2 Schematic Application of a Suspension Operation ($\sim$)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Core Attribute</th>
<th>Cont. Attribute</th>
<th>Prn</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td></td>
<td>\texttt{K}</td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td></td>
<td>\texttt{K}</td>
<td></td>
</tr>
</tbody>
</table>

For example, in 3.1.2 (line1), \textit{Perhaps} and \textit{Fido} cannot be connected because there is no semantic relation between them (suspension). They wait at the now front until the hear mode operations $\text{SBJ} \times \text{PRD}$ (3.1.4) and $\text{ADV} \times \text{FV}$ (3.1.5) are simultaneously activated by the arrival of \textit{is}, which matches the core attribute of their second input pattern. Finding different proplets at the now front matching their respective first input pattern, $\text{SBJ} \times \text{PRD}$ connects \textit{Fido} and \textit{is}, and $\text{ADV} \times \text{FV}$ connects \textit{Perhaps} and \textit{is} (3.1.2, lines 2a and 2b). A suspension may be compensated by cross-copying or by absorption.

In contradistinction to cross-copying ($\times$) and suspension ($\sim$), an absorption operation ($\cup$) consists of two input, but of only one output pattern:

### 6.1.3 Schematic Application of an Absorption Operation ($\cup$)

<table>
<thead>
<tr>
<th>Pattern Level</th>
<th>Core Attribute</th>
<th>Cont. Attribute</th>
<th>Prn</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>$\texttt{X}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$\texttt{Y}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Absorption combines a function proplet and a content proplet into one. For example, $\text{DET} \cup \text{CN}$ in 3.2.6 absorbs the content proplet \textit{bone} into the function
proplet $n_I$ representing the determiner the, while $S \cup IP$ in [2.1.3] absorbs the function proplet *sleep*. If a substitution variable to be replaced by an absorption operation, e.g. $n_2$ in [2.4.2] line 6, occurs more than once at the current now front, all its instances are automatically replaced (simultaneous substitution).

All hear mode operations take place at the now front of the agent’s memory. The trigger for activating a certain hear mode operation is the current next word matching its second input pattern. An activated hear mode operation applies if it finds a proplet matching its first input pattern at the now front and replaces the input with the output. The content resulting from successful hear mode operations is cleared from the now front in regular intervals by moving the now front into free memory space, leaving current content behind to join the permanent sediment of the database (loom-like now front clearance).

The names of the hear mode operations use the following terms to specify the grammatical role of the input patterns:

6.1.4 Proplet Kinds Specified by DBS.Hear Operation Names

1. ADN = adnominal, adj proplet
2. ADV = adverb, adj proplet
3. AUX = auxiliary, verb proplet
4. BP = bare preposition, noun proplet
5. CN = common noun, noun proplet
6. CNJ = coordinating conjunction, noun, verb or adj proplet
7. DET = determiner, noun proplet
8. do = do, auxiliary verb
9. FV = finite verb, verb proplet
10. INF = infinitive, verb proplet
11. IP = interpunctuation, verb proplet
12. N = noun, noun proplet
13. NFV = nonfinite verb, verb proplet
14. NOM = nominative, noun proplet
15. NP = noun phrase, noun proplet
16. OBJ = object, noun proplet
17. PRD = predicate, verb proplet
18. PREP = preposition, noun proplet
19. S = sentence, verb proplet
20. SBJ = subject, noun proplet
21. sub = subordinating conjunction, noun, verb, or adj proplet
22. WH = wh-proplet, verb proplet

A proplet is called a noun proplet if its core attribute is noun, and correspondingly for verb and adj proplets.

To indicate special grammatical conditions, some of the terms are further distinguished by the following diacritics (in alphabetical order):

**6.1.5 Diacritics used by DBS.Hear operation patterns**

1. PRD\(^o\), predicate of object gapping (5.4.4, 5.4.8, 5.4.10)
2. PRD\(^p\), shared item of predicate gapping (5.3.6, 5.3.11)
3. PREP\(^a\), modifier as obligatory argument (4.1.6)
4. PREP\(^m\), optional modifier (3.2.7)
5. SBJ\(^p\), shared item of subject gapping (5.2.6, 5.2.11)
6. sub\(^\text{clause}\), suspended subordinating conjunction (2.5.3, 2.5.3, 3.5.3, 3.5.5, 5.5.8, 5.5.10, 5.5.12, 5.5.15)
7. sub\(^\text{that}\), subject and object clause conjunction (2.5.6, 2.6.4, 5.5.7, 5.5.11)
8. sub\(^\text{when}\), adverbial clause conjunction (3.5.8)
9. sub\(^\text{who}\), adnominal clause conjunction, subject function (3.3.4, 5.6.6, 5.6.10)
10. sub\(^\text{whom}\), adnominal clause conjunction, object function (3.4.5, 3.4.7)

**6.2 Ordering the Operations of DBS.Hear**

The hear mode analysis of the 24 constructions in Sects. 2.1–5.6 required the definition of 43 operations. Presenting them as a DBS.Hear grammar raises the question of whether they should be ordered and if so, how.

From a programming point of view, there is no need to store a set of hear mode operations in a certain order. An index (inverted file, random access) may be used to relate any proplet structure to any subset of operations. Conceptually, however, a linguistically meaningful order is important for upscaling and debugging.

The first linguistic principle of ordering the DBS.Hear operations is sorting operations with the same core attribute in their second input pattern into relevant subsets. This is because DBS.Hear operations are activated by the current next word. Thus, if the core attribute of a current next word is verb, for example, the only operations with a chance of being successful are those which have the core attribute verb in their second input pattern.

---

1 Coordinates approach, CLaTR Sect. 4.1.
Because there are only three different core attributes in DBS, namely verb, noun, and adj, the relevant subsets turn out to be fairly large. For example, in DBS.Hear \[6.3.1\], the subset verb contains 21 and the subset noun 19 operations. With upscaling, the subsets will become even bigger. It is therefore useful to divide the relevant subsets according to additional principles.

The second principle for sorting DBS.Hear operations within a relevant subset is by the same connective, i.e. \(\times\), \(\sim\), or \(\cup\). This is meaningful for debugging and upscaling because two operations with different connectives could never be combined. Operations in the same relevant subset with the same connective, in contrast, raise the possibility of being generalized into one.

The third and last principle is the distinction between DBS.Hear operations without and with diacritics \[6.1.5\]. The purpose of the diacritics is to limit operations to certain special grammatical situations, like the subject being the shared item of a gapping construction. Operations without diacritics, in contrast, are of a more general nature.

In accordance with these three principles, the 43 operations developed for parsing the 24 constructions of the example set \[1.6.1\] are ordered as follows:

### 6.2.1 ORDERING THE OPERATIONS OF DBS.Hear

1. share second core attribute verb 1-21
   1.1 share connective \(\times\) 1-11
   1.1.1 without diacritics 1-3
   1.1.2 with diacritics 4-11
   1.2 share connective \(\sim\) 12-13
   1.2.1 without diacritics 12-13
   1.2.2 with diacritics
   1.3 share connective \(\cup\) 14-21
   1.3.1 without diacritics 14-18
   1.3.2 with diacritics 19-21

2. share second core attribute noun 22-40
   2.1 share connective \(\times\) 22-31
   2.1.1 without diacritics 22-24
   2.1.2 with diacritics 25-31
   2.2 share connective \(\sim\) 32-36
   2.2.1 without diacritics 32-33
   2.2.2 with diacritics 34-36
   2.3 share connective \(\cup\) 37-40
   2.3.1 without diacritics 37-40
   2.3.2 with diacritics
6. The DBS Grammars

3. share second core attribute adj 41-43

3.1 share connective × 41-43
3.1.1 without diacritics 41-43
3.1.2 with diacritics
3.2 share connective ∼
3.2.1 without diacritics
3.2.2 with diacritics
3.3 share connective ∪
3.3.1 without diacritics
3.3.2 with diacritics

The reason for the low number of adj operations is our choice of the test list 1.6.1. A sample set dedicated to the empirical study of modifiers would require additional operations which would fill the currently empty slots.

6.3 Definition of DBS.Hear

This section presents the DBS.Hear grammar which resulted from the linguistic analysis of the test list in Sect. 1.6. It consists of 43 operations which are presented in the form of declarative specifications for a computational implementation. The operations are ordered in accordance with the principles described in the previous section and specified explicitly in 6.2.1.

Each operation is presented as an entry which consists of (i) a consecutive item number, (ii) the operation name, (iii) the list of applications in the interpretation of the sample set 1.6.1 and (iv) the definition of the operation, including possible instructions and conditions.

6.3.1 Listing the Operations of DBS.Hear

1. **SBJ × PRD**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun: α</td>
<td>cat: NP</td>
</tr>
<tr>
<td>cat: NP</td>
<td>fnc: prn: K</td>
</tr>
<tr>
<td>Agreement conditions as in 2.1.2</td>
<td>⇒</td>
</tr>
<tr>
<td>verb: β</td>
<td>cat: NP</td>
</tr>
<tr>
<td>cat: NP</td>
<td>fnc: prn: K</td>
</tr>
<tr>
<td>arg: α</td>
<td>prn: K</td>
</tr>
</tbody>
</table>

---

2 In the online version, automatic cross-referencing may be used. Thus, clicking on 2.1.2, for example, transports the reader automatically to the application of SBJ × PRD in Chap. 2, Sect. 2.1. Clicking go and then back in the top tool bar of Adobe pdf gets the reader back to the present position.
2. ADV×FV

\[ \begin{align*}
\text{adj: } & \alpha \\
\text{cat: } & \text{ADV} \\
\text{mdd:} & \\
\text{prn: } & K
\end{align*} \quad \Rightarrow \quad
\begin{align*}
\text{adj: } & \alpha \\
\text{cat: } & \text{ADV} \\
\text{mdd:} & \gamma \\
\text{prn: } & K
\end{align*} \]

ADV ∈ \{adv, adnv\}

3. WH×FV

\[ \begin{align*}
\text{noun: } & \text{WH}_n \\
\text{cat: } & \text{obq} \\
\text{fnc:} & \\
\text{prn: } & L\&
\end{align*} \quad \Rightarrow \quad
\begin{align*}
\text{noun: } & \text{WH}_n \\
\text{cat: } & \text{obq} \\
\text{fnc:} & \alpha \\
\text{prn: } & L\&
\end{align*} \]

4. SBJ×PRD

\[ \begin{align*}
\text{noun: } & \alpha \\
\text{cat: } & \text{NP} \\
\text{fnc:} & X \\
\text{prn: } & K
\end{align*} \quad \Rightarrow \quad
\begin{align*}
\text{noun: } & \alpha \\
\text{cat: } & \#\text{NP} \ Y \\
\text{fnc:} & \alpha \\
\text{prn: } & K
\end{align*} \]

Agreement conditions the same as in 2.1.2.

5. subthat×PRD

\[ \begin{align*}
\text{verb: } & \alpha \\
\text{cat: } & \#X’ \ v \\
\text{sem: } & \text{that} \ Y \\
\text{fnc:} & \\
\text{prn: } & K
\end{align*} \quad \Rightarrow \quad
\begin{align*}
\text{verb: } & \alpha \\
\text{cat: } & \#\text{NP}’ \ Y \ v \\
\text{sem: } & \text{that} \ Y \\
\text{fnc:} & (\beta K+1) \\
\text{prn: } & K+1
\end{align*} \]

NP’ ∈ \{ns3’, n-s3’, n’\}

6. subwhen×PRD

\[ \begin{align*}
\text{verb: } & \alpha \\
\text{cat: } & \#X’ \ v \\
\text{sem: } & \text{that} \ Y \\
\text{fnc:} & \\
\text{prn: } & K
\end{align*} \quad \Rightarrow \quad
\begin{align*}
\text{verb: } & \alpha \\
\text{cat: } & \#\text{NP}’ \ Y \ v \\
\text{sem: } & \text{that} \ Y \\
\text{fnc:} & \alpha \\
\text{prn: } & K+1
\end{align*} \]

7. PRD×subthat

\[ \begin{align*}
\text{verb: } & \alpha \\
\text{mdd:} & \\
\text{prn: } & K
\end{align*} \quad \Rightarrow \quad
\begin{align*}
\text{verb: } & \alpha \\
\text{mdd: } & (\beta K+1) \\
\text{prn: } & K+1
\end{align*} \]

\[ \begin{align*}
\text{verb: } & \beta \\
\text{cat: } & \#X’ \ #a’ \ v \\
\text{sem: } & \text{that} \ Y \\
\text{fnc:} & (\beta K+1) \\
\text{prn: } & K+1
\end{align*} \]
8. **PRD** × **sub**

\[
\begin{align*}
\verb: & \beta \\
\text{cat}: & \#NP' \ a' \ V \\
\text{arg}: & \gamma \\
\text{prn}: & K
\end{align*}
\]

\[
\begin{align*}
\verb: & V_n \\
\text{sem}: & \alpha \\
\text{fnc}: & \beta (V_n \ K+1) \\
\text{prn}: & K
\end{align*}
\]

\[
\begin{align*}
\verb: & \beta \\
\text{cat}: & \#NP' \ #a' \ v \\
\text{arg}: & \gamma (V_n \ K+1) \\
\text{prn}: & K
\end{align*}
\]

Agreement conditions as in 2.6.4

9. **CN** × **sub**

\[
\begin{align*}
\verb: & \beta \\
\text{cat}: & \#X' \ N' \ v \\
\text{arg}: & Y \\
\text{prn}: & K
\end{align*}
\]

\[
\begin{align*}
\verb: & V_n \\
\text{sem}: & \text{that} \\
\text{fnc}: & \beta K \\
\text{prn}: & K+1
\end{align*}
\]

Agreement conditions as in 2.6.4

10. **CN** × **sub**

\[
\begin{align*}
\verb: & \beta \\
\text{cat}: & \text{NP} \\
\text{arg}: & \emptyset \\
\text{prn}: & K
\end{align*}
\]

\[
\begin{align*}
\verb: & \alpha \\
\text{cat}: & \text{NP} \\
\text{arg}: & \emptyset \\
\text{prn}: & K
\end{align*}
\]

\[
\begin{align*}
\verb: & V_n \\
\text{cat}: & \{NP'\} \\
\text{arg}: & \emptyset \\
\text{prn}: & K+1
\end{align*}
\]

Agreement conditions as in 3.3.4

11. **PRD** × **INF**

\[
\begin{align*}
\verb: & \alpha \\
\text{cat}: & \#NOM' \ a' \ v \\
\text{arg}: & \beta \\
\text{prn}: & K
\end{align*}
\]

\[
\begin{align*}
\verb: & v_1 \\
\text{cat}: & \text{inf} \\
\text{fnc}: & \alpha \\
\text{arg}: & \beta \\
\text{prn}: & K
\end{align*}
\]

\[
\begin{align*}
\verb: & \alpha \\
\text{cat}: & \#NOM' \ #a' \ v \\
\text{arg}: & \beta \ v_1 \\
\text{prn}: & K
\end{align*}
\]

Agreement conditions as in 4.4.4
\[ \alpha \epsilon \{ \text{begin, can afford, choose, decide, expect, forget, learn, like, manage, need, offer, plan, prepare, refuse, start, try, want} \} \]

12. **V~CNJ**

(5.4.6)

\[
\begin{align*}
\verb: \alpha & \quad \verb: \alpha \\
\text{sem: } \beta & \quad \text{sem: } \beta \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

\[ \Rightarrow \]

\[
\begin{align*}
\verb: \alpha & \quad \verb: \alpha \\
\text{sem: } \beta & \quad \text{sem: } \beta \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

\[ \beta \epsilon \{ \text{and, or} \} \]

13. **WH~AUX**

(5.5.4)

\[
\begin{align*}
\text{noun: } V_n & \quad \verb: \alpha \\
\text{cat: } \text{obq} & \quad \text{cat: } \text{NOM AUX v} \\
\text{prn: } K & \quad \text{prn: } \text{obq}
\end{align*}
\]

\[ \Rightarrow \]

\[
\begin{align*}
\text{noun: } V_n & \quad \verb: \alpha \\
\text{cat: } \text{obq} & \quad \text{cat: } \text{NOM AUX v} \\
\text{prn: } K & \quad \text{prn: } \text{K&} \\
\text{prn: } K & \quad \text{prn: } K+1
\end{align*}
\]

14. **CNJ\cup V**

(5.2.10)

\[
\begin{align*}
\verb: V_n & \quad \verb: \alpha \\
\text{sem: } \beta & \quad \text{sem: } \beta \\
\text{prn: } K & \quad \text{prn: } K \\
\end{align*}
\]

\[ \beta \epsilon \{ \text{and, or} \} \]

15. **AUX\cup NFV**

(2.2.6, 2.2.7, 3.1.7)

\[
\begin{align*}
\verb: V_n & \quad \verb: \alpha \\
\text{cat: } \#W' AUX' \text{ VT} & \quad \text{cat: } \text{Y AUX} \\
\text{sem: } X & \quad \text{sem: } Z \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

\[ \Rightarrow \]

\[
\begin{align*}
\verb: \alpha & \quad \verb: \alpha \\
\text{cat: } \#W' \#AUX' \text{ Y VT} & \quad \text{cat: } \#W' \#AUX' \text{ Y VT} \\
\text{sem: } X Z & \quad \text{sem: } X Z \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

AUX \epsilon \{ \text{be, hv, do}\}. VT \epsilon \{ \text{v, vi} \}.

16. **do\cup INF**

(5.5.6)

\[
\begin{align*}
\verb: V_n & \quad \verb: \beta \\
\text{cat: } \#\text{NOM' do' v} & \quad \text{cat: } \#\text{NOM' do' v} \\
\text{sem: } X & \quad \text{sem: } \text{pres} \\
\text{arg: } \alpha & \quad \text{arg: } \alpha \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

\[ \Rightarrow \]

\[
\begin{align*}
\verb: \beta & \quad \verb: \beta \\
\text{cat: } \#\text{NOM' X' vi} & \quad \text{cat: } \#\text{NOM' X' vi} \\
\text{sem: } \text{do } X & \quad \text{sem: } \text{do } X \\
\text{arg: } \alpha & \quad \text{arg: } \alpha \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]
17. **INF ∪ V**

\[
\begin{align*}
\verb|v_1| & : \verb|\alpha| \\
\verb|\textrm{cat}| & : \verb|\textrm{inf}| \\
\verb|\arg| & : \beta \\
\verb|\textrm{prn}| & : K
\end{align*}
\Rightarrow
\begin{align*}
\verb|\alpha| & : \verb|\beta| \\
\verb|\textrm{cat}| & : \verb|\textrm{NP'} \ X \ v| \\
\verb|\arg| & : \gamma \\
\verb|\textrm{prn}| & : K
\end{align*}
\]

18. **S ∪ IP**

\[
\begin{align*}
\verb|\alpha| & : \verb|\beta| \\
\verb|\textrm{cat}| & : \verb|\textrm{VT}| \\
\verb|\textrm{sem}| & : \alpha \\
\verb|\arg| & : \gamma \\
\verb|\textrm{prn}| & : K
\end{align*}
\Rightarrow
\begin{align*}
\verb|\beta| & : \verb|\alpha| \\
\verb|\textrm{cat}| & : \verb|\textrm{NP'} \ X \ v| \\
\verb|\textrm{sem}| & : \gamma \\
\verb|\arg| & : \gamma \\
\verb|\textrm{prn}| & : K
\end{align*}
\]

Agreement conditions as in 2.1.3.

19. **sub clause ∪ PRD**

\[
\begin{align*}
\verb|\alpha| & : \verb|\beta| \\
\verb|\textrm{cat}| & : \verb|\textrm{NP'} | \\
\verb|\textrm{sem}| & : \alpha \\
\verb|\arg| & : \emptyset \\
\verb|\textrm{mdd}| & : (\gamma K^{-1}) \\
\verb|\textrm{prn}| & : K
\end{align*}
\Rightarrow
\begin{align*}
\verb|\beta| & : \verb|\alpha| \\
\verb|\textrm{cat}| & : \verb|\textrm{NP'} | \\
\verb|\textrm{sem}| & : \gamma \\
\verb|\arg| & : \emptyset \\
\verb|\textrm{mdd}| & : (\gamma K^{-1}) \\
\verb|\textrm{prn}| & : K
\end{align*}
\]

\(\alpha \in \{\text{that, when, where, how, why}\}\)

20. **sub who ∪ PRD**

\[
\begin{align*}
\verb|\alpha| & : \verb|\beta| \\
\verb|\textrm{cat}| & : \verb|\textrm{NP'} | \\
\verb|\textrm{sem}| & : \alpha \\
\verb|\arg| & : \emptyset \\
\verb|\textrm{mdd}| & : (\gamma K^{-1}) \\
\verb|\textrm{prn}| & : K
\end{align*}
\Rightarrow
\begin{align*}
\verb|\beta| & : \verb|\alpha| \\
\verb|\textrm{cat}| & : \verb|\textrm{NP'} | \\
\verb|\textrm{sem}| & : \gamma \\
\verb|\arg| & : \emptyset \\
\verb|\textrm{mdd}| & : (\gamma K^{-1}) \\
\verb|\textrm{prn}| & : K
\end{align*}
\]

\(\alpha \in \{\text{who, which}\}\). Agreement conditions as in 3.3.5.

21. **sub whom ∪ PRD**

\[
\begin{align*}
\verb|\alpha| & : \verb|\beta| \\
\verb|\textrm{cat}| & : \verb|\textrm{NP'} | \\
\verb|\textrm{sem}| & : \alpha \\
\verb|\arg| & : \emptyset \\
\verb|\textrm{mdd}| & : (\gamma K^{-1}) \\
\verb|\textrm{prn}| & : K
\end{align*}
\]
6.3 Definition of DBS.Hear

\[
\begin{align*}
\text{verb: } & V_n \\
\text{cat: } & \alpha \\
\text{sem: } & X \\
\text{arg: } & Y \\
\text{mdd: } & (\gamma K-1) \\
\text{prn: } & K \\
\alpha & \in \{\text{who(m), which}\}
\end{align*}
\]

22. PRD\times OBJ

[2.3.4, 2.4.6, 3.2.5, 3.3.6, 4.1.4, 4.2.5, 4.3.4, 4.4.6, 4.5.4, 5.2.4, 5.2.7, 5.2.12, 5.3.4, 5.3.7, 5.3.12, 5.6.4, 5.6.8, 5.6.12]

\[
\begin{align*}
\text{verb: } & \beta \\
\text{cat: } & \#X' N' Y \\
\text{arg: } & Z \\
\text{prn: } & K \\
\Rightarrow \\
\text{noun: } & \alpha \\
\text{mdr: } & \text{CN' N} \\
\text{fnc: } & \beta \\
\text{prn: } & K \\
\end{align*}
\]

Agreement conditions as in 2.3.4

23. do\times NOM

[5.5.5]

\[
\begin{align*}
\text{verb: } & V_n \\
\text{cat: } & \text{NOM'} do' v \\
\text{arg: } & \text{NOM} \\
\text{prn: } & K \\
\Rightarrow \\
\text{noun: } & \alpha \\
\text{mdr: } & \text{CN} \\
\text{fnc: } & V_n \\
\text{prn: } & K \\
\end{align*}
\]

24. N\times PREP

[5.1.3, 5.1.6, 5.1.9]

\[
\begin{align*}
\text{noun: } & \alpha \\
\text{mdr: } & \text{adnv} \\
\text{prn: } & K \\
\Rightarrow \\
\text{noun: } & \alpha \\
\text{mdr: } & \text{N_n} \\
\text{prn: } & K \\
\end{align*}
\]

25. PRD\times PREP

[3.2.7]

\[
\begin{align*}
\text{verb: } & \alpha \\
\text{mdr: } & \text{adnv} \\
\text{prn: } & K \\
\Rightarrow \\
\text{noun: } & \alpha \\
\text{mdr: } & \text{N_n} \\
\text{prn: } & K \\
\end{align*}
\]

26. PRD\times PREP

[4.1.6]
298 6. The DBS Grammars

\[
\begin{align*}
\text{verb: } \alpha & \quad \text{noun: } N_n \\
\text{cat: } \#X \text{ mdr' } Y & \quad \text{cat: } \text{adv} \\
\text{mdr: } & \quad \text{mdd: } \\
\text{prn: } K & \quad \text{prn: } K \\
\Rightarrow & \\
\text{verb: } \alpha & \quad \text{noun: } N_n \\
\text{cat: } \#X \text{ mdr' } Y & \quad \text{cat: } \text{adv} \\
\text{mdr: } & \quad \text{mdd: } \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

27. PRD^p × OBJ

(5.4.10)

\[
\begin{align*}
\text{verb: } \beta & \quad \text{noun: } \alpha \\
\text{cat: } \#X' \text{ N' v} & \quad \text{cat: } \text{CN' N} \\
\text{arg: } Z & \quad \text{fnc: } \\
\text{prn: } K & \quad \text{prn: } K \\
\Rightarrow & \\
\text{verb: } \beta & \quad \text{noun: } \alpha \\
\text{cat: } \#X' \text{ N' v} & \quad \text{cat: } \text{CN' N} \\
\text{arg: } Z & \quad \text{fnc: } V_o \beta \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

Replace all occurrences of n_c with α by simultaneous substitution.

28. N × N

(3.6.6)

\[
\begin{align*}
\text{noun: } \alpha & \quad \text{noun: } \beta \\
\text{nc: } & \quad \text{nc: } \\
\text{pc: } \gamma & \quad \text{pc: } \gamma \\
\text{prn: } K & \quad \text{prn: } K \\
\Rightarrow & \\
\text{noun: } \alpha & \quad \text{noun: } \beta \\
\text{nc: } & \quad \text{nc: } \\
\text{pc: } \gamma & \quad \text{pc: } \gamma \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

The sem slot of α may not contain a conjunction.

29. N_1 × N

(3.6.3)

\[
\begin{align*}
\text{noun: } \alpha & \quad \text{fnc: } \\
\text{nc: } & \quad \text{nc: } \\
\text{pc: } & \quad \text{pc: } \\
\text{prn: } K & \quad \text{prn: } K \\
\Rightarrow & \\
\text{noun: } \alpha & \quad \text{fnc: } \\
\text{nc: } & \quad \text{nc: } \\
\text{pc: } & \quad \text{pc: } \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

30. PRD^p × SBJ

(5.3.6, 5.3.11)

\[
\begin{align*}
\text{verb: } \beta & \quad \text{noun: } \alpha \\
\text{cat: } \#NP' \text{ #X v} & \quad \text{cat: } \text{NP} \\
\text{arg: } Y & \quad \text{fnc: } \\
\text{prn: } K & \quad \text{prn: } K \\
\Rightarrow & \\
\text{verb: } \beta & \quad \text{noun: } \alpha \\
\text{cat: } \#NP' \text{ X v} & \quad \text{cat: } \text{NP} \\
\text{arg: } Y & \quad \text{fnc: } \\
\text{prn: } K & \quad \text{prn: } K
\end{align*}
\]

Agreement conditions as in 4.5.4

31. PRD^p × SBJ

(5.4.4, 5.4.8)
Add the value $\alpha$ to the storage variable $v_o$.

32. \textit{ADV} \sim \textit{NOM}

\textbf{(3.1.3)}

\begin{align*}
\text{adj: } & \alpha \\
\text{cat: } & \text{adv} \\
\text{mdd: } & \\
\text{fnc: } & \\
\text{prn: } & K
\end{align*}

\begin{align*}
\Rightarrow
\begin{align*}
\text{noun: } & \beta \\
\text{sem: } & C \\
\text{prn: } & K
\end{align*}
\end{align*}

$C \in \{\text{and, or}\}$

33. \textit{N} \sim \textit{CNJ}

\textbf{(3.6.4, 5.2.9)}

\begin{align*}
\text{noun: } & \alpha \\
\text{sem: } & X \\
\text{prn: } & K
\end{align*}

\begin{align*}
\Rightarrow
\begin{align*}
\text{noun: } & C \\
\text{sem: } & C \\
\text{prn: } & K
\end{align*}
\end{align*}

$C \in \{\text{and, or}\}$

34. \textit{sub}^{clause} \sim \textit{SBJ}

\textbf{(2.5.3, 2.6.5, 3.5.3, 5.5.8, 5.5.12)}

\begin{align*}
\text{verb: } & V_n \\
\text{sem: } & \alpha \\
\text{prn: } & K
\end{align*}

\begin{align*}
\Rightarrow
\begin{align*}
\text{noun: } & \beta \\
\text{sem: } & \alpha \\
\text{prn: } & K
\end{align*}
\end{align*}

$\alpha \in \{\text{that, when, where, how, why}\}$

35. \textit{sub}^{whom} \sim \textit{SBJ}

\textbf{(3.4.5)}

\begin{align*}
\text{verb: } & V_n \\
\text{sem: } & \alpha \\
\text{prn: } & K
\end{align*}

\begin{align*}
\Rightarrow
\begin{align*}
\text{noun: } & \beta \\
\text{sem: } & \alpha \\
\text{prn: } & K
\end{align*}
\end{align*}

$\alpha \in \{\text{whom, which}\}$

36. \textit{sub}^{when} \sim \textit{NOM}

\textbf{(3.5.6)}

\begin{align*}
\text{verb: } & \beta \\
\text{cat: } & \#X' \#N' v \\
\text{sem: } & \alpha Z \\
\text{prn: } & K
\end{align*}

\begin{align*}
\Rightarrow
\begin{align*}
\text{noun: } & \gamma \\
\text{cat: } & \#X' \#N' v \\
\text{sem: } & \alpha Z \\
\text{prn: } & K
\end{align*}
\end{align*}

$\alpha \in \{\text{when, where, how, why}\}, \ NP \in \{\text{snp, pnp, s1, s3, p1, sp2, p3}\}.$
37. $\text{DET} \cup \text{CN}$

```
\begin{align*}
\text{noun: } N_n & \quad \text{cat: } X \text{ CN' NP} \\
\text{sem: } Y & \quad \text{prn: } K \\
\end{align*}
\Rightarrow
\begin{align*}
\text{noun: } \alpha & \quad \text{cat: } \text{CN} \\
\text{sem: } Z & \quad \text{prn: } K \\
\end{align*}
```

Delete any $\theta a+$ marker.

Agreement conditions as in 2.2.4.

38. $\text{PREP} \cup \text{NP}$

```
\begin{align*}
\text{noun: } N_n & \quad \text{cat: adnv} \\
\text{sem: } X & \quad \text{prn: } K \\
\end{align*}
\Rightarrow
\begin{align*}
\text{noun: } \alpha & \quad \text{cat: CN' NP} \\
\text{sem: } Z & \quad \text{prn: } K \\
\end{align*}
```

39. $\text{CNJ} \cup \text{UN}$

```
\begin{align*}
\text{noun: } n_1 & \quad \text{cat: } \beta \\
\text{sem: } \beta & \quad \text{prn: } K \\
\beta & \in \{\text{and, or}\} \\
\end{align*}
\Rightarrow
\begin{align*}
\text{noun: } \alpha & \quad \text{cat: } \beta X \\
\text{sem: } \beta X & \quad \text{prn: } K \\
\end{align*}
```

40. $\text{PRD} \cup \text{BP}$

```
\begin{align*}
\text{verb: } \alpha & \quad \text{cat: } \#X \beta' v \\
\text{cat: adnv} & \text{sem: } Y \\
\text{prn: } K & \quad \text{prn: } \beta \\
\beta & \in \{\text{in, out, up,} \ldots\} \\
\end{align*}
\Rightarrow
\begin{align*}
\text{verb: } \alpha & \quad \text{cat: } \#X \#\beta' v \\
\text{cat: adnv} & \text{sem: } \beta Y \\
\text{prn: } K & \quad \text{prn: } K \\
\end{align*}
```

41. $\text{DET} \times \text{ADN}$

```
\begin{align*}
\text{verb: } \alpha & \quad \text{cat: } \#X \beta' v \\
\text{cat: adnv} & \text{sem: } Y \\
\text{prn: } K & \quad \text{prn: } \beta \\
\beta & \in \{\text{in, out, up,} \ldots\} \\
\end{align*}
\Rightarrow
\begin{align*}
\text{verb: } \alpha & \quad \text{cat: } \#X \#\beta' v \\
\text{cat: adnv} & \text{sem: } \beta Y \\
\text{prn: } K & \quad \text{prn: } K \\
\end{align*}
```

\[3.6.5, 5.6.9\]
6.4 Four Operation Formats of DBS.Speak

While the hear mode is driven by a sequence of incoming surfaces, the selective activation of the think mode is driven by an autonomous navigation along the semantic relations of structure which connect the proplets of a content. These relations are characterized graphically by different kinds of lines, called *edges* in graph theory. Using a linear notation, the lines are also used in the names of navigation operations to characterize the nature of the semantic relation traversed. More specifically, the variants of the *functor argument* relation are shown as (i) subject/predicate, (ii) object\predicate, and (iii) modifier|modified, while *coordination* is shown as (iv) conjunct−conjunct.

A speak mode operation is essentially a think mode operation\(^3\) with an additional language-dependent lexicalization rule embedded into the *sur* slot of

\(^3\) In line with the DBS laboratory set-up, the think mode operations used here are limited to the selective activation of content derived by DBS.Hear, leaving the inferencing aspect of the think mode aside.
the goal pattern. The lexicalization rules take the core value and the grammatical properties of their proplet as input and produce zero, one, or more language-dependent surfaces which are passed to the agent’s action components for realization. Here, the definitions of DBS.Speak operations concentrate on the think mode navigation aspect, leaving the lexicalization aspect to informal remarks in Chaps. 2–5.

To indicate the direction of a think mode navigation, arrowheads are added to the lines, i.e. / (downward) and / (upward) for subject/predicate, \ (downward) and \ (upward) for object\predicate, ↓ (downward) and ↑ (upward) for modifier|modified, and → (left-to-right) and ← (right-to-left) for conjunct—conjunct.

Consider a schematic application of \N:

### 6.4.1 Schematic Application of a V/\N Operation

\[ \begin{array}{c|c|c|c|c|c|c|c} \hline \text{pattern level} & \verb: \alpha & \arg: \beta X & \prn: K & \Rightarrow & \verb: \beta & \fnc: \alpha & \prn: K \\ \hline \text{content level} & \verb: eat & \arg: \#dog bone & \prn: 23 \\ \hline \end{array} \]

The feature \[\text{arg: } \beta X\] of the input pattern picks the initial \verb value for the continuation, i.e. the subject, as its goal proplet. The / relation requires altogether four operations, (i) intrapropositionally for elementary (e.g. 2.1.6) and phrasal (e.g. 2.2.13) subjects, and (ii) extrapropositionally for clausal subjects (2.5.11), each in both directions, i.e. (1) \N, (2) \N, (3) \N, (4) \N.

The next example shows a schematic application of \N:

### 6.4.2 Schematic Application of a V/\N Operation

\[ \begin{array}{c|c|c|c|c|c|c|c} \hline \text{pattern level} & \verb: \alpha & \arg: \#X \beta Y & \prn: K & \Rightarrow & \verb: \beta & \fnc: \#\alpha & \prn: K \\ \hline \text{content level} & \verb: eat & \arg: \#\dog bone & \prn: 24 \\ \hline \end{array} \]

\[\text{\footnotesize For explicitly defined lexicalization rules see NLC Sects. 12.4–12.6.}\]
The feature $[\text{arg: } \#X \beta Y]$ of the input pattern picks a noninitial arg value for the continuation, i.e. the first unmarked object, as its goal proplet. The \ relation requires altogether four operations, (i) intrapropositionally for elementary (e.g. 2.3.10) and phrasal (e.g. 2.4.18) objects, and (ii) extrapropositionally for clausal objects (2.6.13), each in both directions, i.e. (5) $V\backslash N$, (6) $N\backslash V$, (7) $V\backslash V$, (8) $V\downarrow V$.

The modifier\ modified relation is of two kinds, adnominal and adverbial. An application of adnominal $\text{A}\uparrow\text{N}$ may be shown schematically as follows:

### 6.4.3 Schematic Application of a $\text{A}\uparrow\text{N}$ Operation

\[
\begin{array}{c|c}
\text{pattern level} & \text{content level} \\
\hline
\begin{array}{c}
\text{adj: } \alpha \\
\text{mdd: } \beta \\
\text{prn: K}
\end{array} & \begin{array}{c}
\text{adj: big} \\
\text{mdd: dog} \\
\text{prn: 25}
\end{array}
\\
\uparrow & \uparrow \\
\text{condition} & \\
\hline
\begin{array}{c}
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: K}
\end{array} & \begin{array}{c}
\text{noun: dog} \\
\text{mdd: big} \\
\text{prn: 25}
\end{array}
\\
\downarrow & \downarrow \\
\text{level} & \\
\hline
\begin{array}{c}
\text{nc: pc: } \alpha \\
\text{prn: K}
\end{array} & \begin{array}{c}
\text{nc: woman} \\
\text{pc: man} \\
\text{prn: 26}
\end{array}
\end{array}
\]

The $[\text{mdd: } \beta]$ pattern of the modifier $\alpha$ together with the prn value specify the address of the modified. The $|$ relation requires altogether eight operations, (i) adnominal intrapropositional for elementary (e.g. 2.4.13) and phrasal (e.g. 3.2.1) modifiers, (ii) adnominal extrapropositional for clausal (e.g. 3.3.12) modifiers, (iii) adverbial intrapropositional for elementary (e.g. 3.1.13) and phrasal (e.g. 3.1.17) modifiers, and (iv) adverbial extrapropositional for clausal (e.g. 3.5.12) modifiers, each in both directions, i.e. (9) $\text{N}\downarrow\text{A}$, (10) $\text{A}\uparrow\text{N}$, (11) $\text{V}\downarrow\text{A}$, (12) $\text{A}\uparrow\text{V}$, (13) $\text{N}\downarrow\text{V}$, (14) $\text{V}\uparrow\text{N}$, (15) $\text{V}\downarrow\text{V}$, (16) $\text{V}\uparrow\text{V}$.

An application of $\text{N}\rightarrow\text{N}$ may be shown schematically as follows:

### 6.4.4 Schematic $\text{N}\rightarrow\text{N}$ Application

\[
\begin{array}{c|c}
\text{pattern level} & \text{content level} \\
\hline
\begin{array}{c}
\text{noun: } \alpha \\
\text{nc: } \beta \\
\text{prn: K}
\end{array} & \begin{array}{c}
\text{noun: man} \\
\text{nc: woman} \\
\text{prn: 26}
\end{array}
\\
\uparrow & \uparrow \\
\text{condition} & \\
\hline
\begin{array}{c}
\text{noun: } \beta \\
\text{pc: } \alpha \\
\text{prn: K}
\end{array} & \begin{array}{c}
\text{noun: woman} \\
\text{pc: man} \\
\text{prn: 26}
\end{array}
\\
\downarrow & \downarrow \\
\text{level} & \\
\hline
\end{array}
\]

The $[\text{nc: } \beta]$ pattern of the current conjunct $\alpha$ together with the prn value specify the address of the next conjunct. The $-$ relation requires altogether seven operations, (i) intrapropositional for elementary (e.g. 3.6.16) and phrasal
304 6. The DBS Grammars

N→N, V→V, and A→A relations, each in both directions, and (ii) extrapropositional for the clausal V→V relation, which is unidirectional (1.4.4), i.e. (17) N→N, (18) N←N, (19) V→V, (20) V←V, (21) A→A, (22) A←A, and (23) V→V.

Similar to DBS.Hear (6.1.4), DBS.Speak operations use terms specifying the grammatical role, though only five instead of 21:

6.4.5 PROPLET KINDS SPECIFIED BY DBS.SPEAK OPERATION NAMES

1. V = verb proplet
2. N = noun proplet
3. A = adj proplet
4. INF = verb proplet
5. WH = noun proplet

Similar to DBS.Hear (6.1.5), DBS.Speak operations use diacritics, here to differentiate particular traversals:

6.4.6 DIACRITICS USED IN THE NAMES OF DBS.SPEAK OPERATIONS

1. N^s indicates the shared item of subject gapping (s23, s24)
2. VP indicates the shared item of predicate gapping (s25, s26)
3. N^f is used for the realization of a shared object (s27, s28)
4. N^o is used for empty traversals of a shared object (s29, s30)
5. N^i is used for realizing the shared subject (s37)
6. V^f is used for the final clause of an unbounded dependency (s38)
7. V^i is used for realizing the shared predicate (s39)

Based on these distinctions, the number of speak mode operations is increased from 23 to a total of 39, which are listed in Sect. 6.5.

Remains the question of how to order the think mode operations for selective activation in a linguistically meaningful way. While hear mode operations have two input patterns, selective activation operations have only one. While hear mode operations are activated by automatic word form recognition matching their second input pattern, selective activation operations use the goal proplet of application n as the start proplet of application n+1 (Continuity Condition, NLC 3.6.5).

Following the ordering of DBS.Hear operations in 6.3, we could divide the speak mode operations into activation sets, defined as sharing the core attribute

---

5 The subscript \( \_x \) indicates extrapropositional use.
of the first pattern. These could in turn be divided according to the connectives /, \, |, and −, and these divided further according to the absence vs. presence of diacritics.

However, because the traversal of a semantic relation of structure is usually traversed in both directions, it is linguistically more instructive to order the DBS.Speak operations into pairs of opposite direction. For example, \ V / N (downward) and N / V (upward) use the same relation and the same patterns for V and N \[1.2.1\]; they differ only in the direction of traversing the semantic relation N / V, as indicated by the arrow heads and the start proplet in initial position \[1.2.2\].

The set of opposite direction pairs is ordered further according to the pair nodes, e.g. N / V vs. V / N, going systematically through the four connectives. Finally, these sets are divided into pairs without and with diacritics.

### 6.5 Definition of DBS.Speak

Similar to the DBS.Hear grammar presented in Sect. \[6.3\] the entries of the following DBS.Speak grammar each consist of (i) a consecutive item number, (ii) the operation name, (iii) the list of applications in the production of the sample set \[1.6.1\] and (iv) the definition of the operation. The ordering is as described in the preceding section. To distinguish the numbers of DBS.Hear operations in Sect. \[6.3\] from the numbers of DBS.Speak operations shown below, the reference number of a hear mode operation is preceded by “h”, as in \[2.2\] (e.g. \[2.3.4\]), in contradistinction to the reference number of a speak mode operation, which is preceded by “s”, as in \[3.2\] (e.g. \[3.6.15\]).

### 6.5.1 Listing the operations of DBS.Speak

1. \[V / N\]

\[\begin{array}{c}
\text{verb: } \alpha \\
\text{arg: } \beta X \\
\text{prn: } K
\end{array}\]

\[\Rightarrow\]

\[\begin{array}{c}
\text{sur: lexnoun}(\beta) \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K
\end{array}\]

#-mark \(\beta\) in the arg slot of proplet \(\alpha\)

\[\text{[1.2.4] 2.1.6 2.2.11 2.3.8 2.4.12 2.5.12 2.6.11 2.6.14 3.1.15 3.2.10}

\[3.2.15 3.3.11 3.4.12 3.5.13 3.5.16 3.6.12 4.1.12 4.2.10 4.3.10}

\[4.4.11 4.5.9 4.6.8 5.1.14 5.4.15 5.4.19 5.4.23 5.5.25 5.5.28 5.5.31}\]

\[5.6.16\]

\[\text{[1.4.4] 1.4.5 1.4.7 5.2.15}\]

An exception is the traversal of extrapropositional conjunct—conjunct relations \[1.4.4\] which is unidirectional. Also, subject \[1.4.5 1.4.7 5.2.15\] and object \[1.4.6 5.4.13\] gapping each have one unidirectional arc.
The instructions of DBS.Speak operations use #-marking to indicate traversed continuation values in the fnc, arg, mdr, nc, and pc slots. This is in contradistinction to #-marking in DBS.hear, which is used to cancel valency positions in the cat slot.

2. N/V

\[
\begin{align*}
\text{noun: } & \beta \\
\text{fnc: } & \alpha \\
\text{mdr: } & Z \\
\text{prn: } & K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } & \text{lexverb}(\hat{\alpha}) \\
\text{verb: } & \alpha \\
\text{arg: } & \#\beta X \\
\text{prn: } & K
\end{align*}
\]

#-mark \(\alpha\) in the fnc slot of proplet \(\beta\)

Z is NIL, or elementary and #-marked

3. V\(\setminus\)N

\[
\begin{align*}
\text{verb: } & \alpha \\
\text{arg: } & \#X \beta Y \\
\text{prn: } & K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } & \text{lexnoun}(\hat{\beta}) \\
\text{noun: } & \beta \\
\text{fnc: } & \alpha \\
\text{prn: } & K
\end{align*}
\]

#-mark \(\beta\) in the arg slot of proplet \(\alpha\)

4. N\(\setminus\)V

\[
\begin{align*}
\text{noun: } & \beta \\
\text{fnc: } & \alpha \\
\text{mdr: } & Z \\
\text{prn: } & K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } & \text{lexverb}(\hat{\beta}) \\
\text{verb: } & \alpha \\
\text{arg: } & \#X \#\beta Y \\
\text{prn: } & K
\end{align*}
\]

Z is #-marked or NIL

5. V\(\setminus\)V

\[
\begin{align*}
\text{verb: } & \alpha \\
\text{arg: } & (\beta K) X \\
\text{prn: } & K+1
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } & \text{lexverb}(\hat{\beta}) \\
\text{verb: } & \beta \\
\text{sem: } & \text{that } Z \\
\text{fnc: } & (\alpha K+1) \\
\text{prn: } & K
\end{align*}
\]
6. \(\frac{\text{V/V}}{\text{2.5.14}}\)  
\[
\begin{align*}
\text{verb: } & \beta \\
\text{sem: } & \text{that } Z \\
\text{arg: } & #X \\
\text{fnc: } & (\alpha K+1) \\
\text{prn: } & K
\end{align*}
\]  
\[
\begin{align*}
\Rightarrow \quad \begin{align*}
\text{sur: } & \text{lexverb}(\hat{\alpha}) \\
\text{verb: } & \alpha \\
\text{arg: } & (\beta K) X \\
\text{prn: } & K+1
\end{align*}
\]

7. \(\frac{\text{V/V}}{\text{2.6.13 5.5.19 5.5.27}}\)  
\[
\begin{align*}
\text{verb: } & \alpha \\
\text{arg: } & X (\beta K+1) \\
\text{prn: } & K
\end{align*}
\]  
\[
\begin{align*}
\Rightarrow \quad \begin{align*}
\text{sur: } & \text{lexverb}(\hat{\beta}) \\
\text{verb: } & \beta \\
\text{sem: } & \text{that } Z \\
\text{fnc: } & (\alpha K) \\
\text{prn: } & K+1
\end{align*}
\]

8. \(\frac{\text{V/V}}{\text{2.6.16 5.5.23 5.5.24 5.5.33 5.5.34}}\)  
\[
\begin{align*}
\text{verb: } & \alpha \\
\text{sem: } & \text{that } Z \\
\text{fnc: } & (\beta K) \\
\text{prn: } & L
\end{align*}
\]  
\[
\begin{align*}
\Rightarrow \quad \begin{align*}
\text{sur: } & \text{lexverb}(\hat{\beta}) \\
\text{verb: } & \beta \\
\text{arg: } & Y (\alpha L) \\
\text{prn: } & K
\end{align*}
\]

9. \(\frac{\text{V/A}}{\text{3.1.13 3.1.17 3.1.19 3.1.21 3.6.18 4.2.14 4.6.10}}\)  
\[
\begin{align*}
\text{verb: } & \alpha \\
\text{mdr: } & #X \beta Y \\
\text{prn: } & K
\end{align*}
\]  
\[
\begin{align*}
\Rightarrow \quad \begin{align*}
\text{sur: } & \text{lexadj}(\hat{\beta}) \\
\text{adj: } & \beta \\
\text{mdd: } & \alpha \\
\text{prn: } & K
\end{align*}
\]  
\(-\text{mark } \beta \text{ in the mdr slot of proplet } \alpha\)

10. \(\frac{\text{A/V}}{\text{3.1.14 3.1.18 3.1.20 3.1.22 3.6.19 4.2.15 4.6.11}}\)  
\[
\begin{align*}
\text{adj: } & \beta \\
\text{mdd: } & \alpha \\
\text{prn: } & K
\end{align*}
\]  
\[
\begin{align*}
\Rightarrow \quad \begin{align*}
\text{sur: } & \text{lexverb}(\hat{\alpha}) \\
\text{verb: } & \alpha \\
\text{mdr: } & X \beta Y \\
\text{prn: } & K
\end{align*}
\]

11. \(\frac{\text{N/A}}{\text{2.2.12 2.4.13 2.4.17}}\)
6. The DBS Grammars

12. A↑N

\[
\begin{bmatrix}
\text{noun: } \beta \\
\text{mdr: } \gamma \\
\text{prn: } K
\end{bmatrix} \Rightarrow
\begin{bmatrix}
\text{sur: } \text{lexadj}(\hat{\gamma}) \\
\text{adj: } \gamma \\
\text{mdd: } \beta \\
\text{prn: } K
\end{bmatrix}
\]

\#-mark \( \gamma \) in the mdr slot of proplet \( \beta \)

13. V↓N

\[
\begin{bmatrix}
\text{verb: } \alpha \\
\text{mdr: } X \, \beta \, Y \\
\text{prn: } K
\end{bmatrix} \Rightarrow
\begin{bmatrix}
\text{sur: } \text{lexnoun}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{bmatrix}
\]

\#-mark \( \beta \) in the arg slot of proplet \( \alpha \)

14. N↑V

\[
\begin{bmatrix}
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{bmatrix} \Rightarrow
\begin{bmatrix}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{mdr: } X \, \beta \, Y \\
\text{prn: } K
\end{bmatrix}
\]

15. N↓V

\[
\begin{bmatrix}
\text{noun: } \alpha \\
\text{mdr: } (\beta \, K+1) \\
\text{prn: } K
\end{bmatrix} \Rightarrow
\begin{bmatrix}
\text{sur: } \text{lexverb}(\hat{\beta}) \\
\text{verb: } \beta \\
\text{mdd: } (\alpha \, K) \\
\text{prn: } K+1
\end{bmatrix}
\]

\#-mark \( \beta \) in the mdr slot of proplet \( \alpha \)

16. V↑N

\[
\begin{bmatrix}
\text{verb: } \beta \\
\text{mdd: } #(\alpha \, K) \\
\text{prn: } K+1
\end{bmatrix} \Rightarrow
\begin{bmatrix}
\text{noun: } \alpha \\
\text{mdr: } #(\beta \, K+1) \\
\text{prn: } K
\end{bmatrix}
\]

17. V↓V

\[
\begin{bmatrix}
\text{verb: } \beta \\
\text{mdr: } X \, \beta \, Y \\
\text{prn: } K+1
\end{bmatrix} \Rightarrow
\begin{bmatrix}
\text{noun: } \alpha \\
\text{mdd: } \alpha \, K \\
\text{prn: } K
\end{bmatrix}
\]\n
\( Z \) is NIL or elementary and #-marked.
6.5 Definition of DBS.Speak

\[
\begin{bmatrix}
\text{verb: } \alpha \\
\text{mdr: } X \beta Y \\
\text{prn: } K
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
\text{sur: lexverb}(\hat{\beta}) \\
\text{verb: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{bmatrix}
\]

#-mark \( \beta \) in the mdr slot of proplet \( \alpha \)

18. \( V \uparrow V \) 
(3.5.15)

\[
\begin{bmatrix}
\text{verb: } \alpha \\
\text{mdr: } X \#\beta Y \\
\text{prn: } K
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
\text{sur: lexverb}(\hat{\beta}) \\
\text{verb: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K+1
\end{bmatrix}
\]

#-mark \( \alpha \) in the mdr slot of proplet \( \beta \)

19. \( N \downarrow N \) 
(5.1.17 5.1.18 5.1.19)

\[
\begin{bmatrix}
\text{noun: } \alpha \\
\text{mdr: } \beta \\
\text{prn: } K
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
\text{sur: lexadj}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{bmatrix}
\]

#-mark \( \beta \) in the arg slot of proplet \( \alpha \)

20. \( N \uparrow N \) 
(5.1.20 5.1.21 5.1.22)

\[
\begin{bmatrix}
\text{noun: } \alpha \\
\text{mdr: } \beta \\
\text{prn: } K
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
\text{sur: lexadj}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{mdd: } \alpha \\
\text{prn: } K
\end{bmatrix}
\]

21. \( N \rightarrow N \) 
(3.6.13 3.6.14)

\[
\begin{bmatrix}
\text{noun: } \alpha \\
\text{nc: } \beta \\
\text{prn: } K
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
\text{sur: lexnoun}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{sem: } \text{CNJ} X \\
\text{pc: } \alpha \\
\text{prn: } K
\end{bmatrix}
\]

22. \( N \leftarrow N \) 
(3.6.15 3.6.16)

\[
\begin{bmatrix}
\text{noun: } \beta \\
\text{pc: } \alpha \\
\text{prn: } K
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
\text{sur: lexnoun}(\hat{\beta}) \\
\text{noun: } \alpha \\
\text{nc: } \beta \\
\text{prn: } K
\end{bmatrix}
\]

23. \( V \searrow N \) 
(5.2.21 5.2.25)
24. NV

\[
\begin{align*}
\text{noun: } \beta & \quad \text{fnc: } X \alpha Y \\
\text{mdr: } Z & \\
\text{prn: } K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{arg: } \beta W \\
\text{prn: } K
\end{align*}
\]

Z is \#-marked or NIL

25. VP/N

\[
\begin{align*}
\text{verb: } \alpha & \quad \text{arg: } #X \beta Y \\
\text{prn: } K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } \text{lexnoun}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{fnc: } \alpha \\
\text{prn: } K
\end{align*}
\]

\#-mark \beta in the arg slot of proplet \alpha

26. N/VP

\[
\begin{align*}
\text{noun: } \beta & \quad \text{fnc: } #X \alpha \beta Y \\
\text{mdr: } Z & \\
\text{prn: } K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{arg: } #X #Y \beta Y \\
\text{prn: } K
\end{align*}
\]

Z is \#-marked or NIL

27. V\setminus N^f

\[
\begin{align*}
\text{verb: } \alpha & \quad \text{arg: } #X \beta \\
\text{prn: } K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } \text{lexnoun}(\hat{\beta}) \\
\text{noun: } \beta \\
\text{fnc: } #Y \alpha \\
\text{prn: } K
\end{align*}
\]

\#-mark \alpha in the fnc slot of proplet \beta

\#-mark \beta in the arg slot of proplet \alpha

28. N^f\setminus V

\[
\begin{align*}
\text{noun: } \beta & \quad \text{fnc: } #X #\alpha \\
\text{mdr: } Z & \\
\text{prn: } K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: } \text{lexverb}(\hat{\alpha}) \\
\text{verb: } \alpha \\
\text{arg: } #W #\beta \\
\text{prn: } K
\end{align*}
\]

\#-mark \alpha in the proplet \beta
Z is #-marked or NIL

29. \(V \backslash N^o\)
   \[5.4.17 \quad 5.4.21\]
   \[
   \begin{align*}
   \text{verb: } \alpha \\
   \text{arg: } \#X \beta Y \\
   \text{prn: } K
   \end{align*}
   \Rightarrow
   \begin{align*}
   \text{sur: lexnoun(}\hat{\beta}\text{)} \\
   \text{noun: } \beta \\
   \text{fnc: } \#Y Z \\
   \text{prn: } K
   \end{align*}
   \]
   #-mark \(\alpha\) in the fnc slot of proplet \(\beta\)

30. \(N^o \backslash V\)
   \[5.4.18 \quad 5.4.22\]
   \[
   \begin{align*}
   \text{noun: } \beta \\
   \text{fnc: } \#X \alpha Y \\
   \text{mdr: } Z \\
   \text{prn: } K
   \end{align*}
   \Rightarrow
   \begin{align*}
   \text{sur: lexverb(}\hat{\alpha}\text{)} \\
   \text{verb: } \alpha \\
   \text{arg: } \beta W \\
   \text{prn: } K
   \end{align*}
   \]
   #-mark \(\alpha\) in the fnc slot of proplet \(\beta\)

31. \(INF \backslash N\)
   \[4.4.14\]
   \[
   \begin{align*}
   \text{verb: } \alpha \\
   \text{arg: } \#X \beta Y \\
   \text{prn: } K
   \end{align*}
   \Rightarrow
   \begin{align*}
   \text{sur: lexnoun(}\hat{\beta}\text{)} \\
   \text{noun: } \beta \\
   \text{fnc: } \#\alpha \\
   \text{prn: } K
   \end{align*}
   \]
   #-mark \(\beta\) in the arg slot of proplet \(\alpha\)

32. \(N \backslash INF\)
   \[4.4.15\]
   \[
   \begin{align*}
   \text{noun: } \beta \\
   \text{fnc: } \#\alpha \\
   \text{prn: } K
   \end{align*}
   \Rightarrow
   \begin{align*}
   \text{sur: lexverb(}\hat{\alpha}\text{)} \\
   \text{verb: } \alpha \\
   \text{arg: } \#X \beta Y \\
   \text{prn: } K
   \end{align*}
   \]
   Z is #-marked or NIL

33. \(V \backslash INF^{try}\)
   \[4.4.13\]
   \[
   \begin{align*}
   \text{verb: } \alpha \\
   \text{arg: } \#\gamma \beta \\
   \text{prn: } K
   \end{align*}
   \Rightarrow
   \begin{align*}
   \text{sur: lexverb(}\hat{\alpha}\text{)} \\
   \text{verb: } \beta \\
   \text{cat: } \#X \text{ inf} \\
   \text{fnc: } \#\alpha \\
   \text{arg: } \#\gamma Y \\
   \text{prn: } K
   \end{align*}
   \]
   #-mark \(\beta\) in the fnc slot of proplet \(\alpha\)
34. **INF/V**

\[
\begin{align*}
\verb: \beta \\
\text{cat}: \#X \text{ inf} \\
\text{arg}: \#Y \\
\text{prn}: K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: lexverb(\hat{\alpha})} \\
\verb: \alpha \\
\text{arg}: \beta \ X \\
\text{prn}: K
\end{align*}
\]

35. **V/WH**

\[
\begin{align*}
\verb: \alpha \\
\text{sem}: that \ X \\
\text{arg}: X \ WH_n \\
\text{prn}: L&
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: lexnoun(\hat{\gamma})} \\
\text{noun: WH}_n \\
\text{fnc}: \alpha \\
\text{prn}: L&
\end{align*}
\]

#-mark WH\_n in the proplet \( \alpha \)

36. **WH/V**

\[
\begin{align*}
\text{noun: WH}_n \\
\text{fnc}: \alpha \\
\text{prn}: L&
\end{align*}
\Rightarrow
\begin{align*}
\verb: \alpha \\
\text{sem}: that \ X \\
\text{arg}: \{ \ X \#WH\_n \} \\
\text{prn}: L&
\end{align*}
\]

37. **V/N\^i**

\[
\begin{align*}
\verb: \alpha \\
\text{arg}: \beta \ X \\
\text{prn}: K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: lexnoun(\hat{\beta})} \\
\text{noun: \beta} \\
\text{fnc}: \alpha \ Z \\
\text{prn}: K
\end{align*}
\]

#-mark \( \beta \) in the arg slot of proplet \( \alpha \)

38. **V/V\^f**

\[
\begin{align*}
\verb: \alpha \\
\text{arg}: X \ (\beta \ L&) \\
\text{prn}: K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: lexverb(\hat{\gamma})} \\
\verb: \beta \\
\text{arg}: Y \ WH_n \\
\text{fnc}: (\alpha \ K) \\
\text{prn}: L&
\end{align*}
\]

39. **N/V\^i**

\[
\begin{align*}
\text{noun: \beta} \\
\text{fnc}: \alpha \\
\text{mdr}: Z \\
\text{prn}: K
\end{align*}
\Rightarrow
\begin{align*}
\text{sur: lexverb(\hat{\alpha})} \\
\verb: \alpha \\
\text{arg}: \#X \#\beta \ Y \\
\text{prn}: K
\end{align*}
\]

#-mark \( \alpha \) in the fnc slot of proplet \( \beta \)
Of the 39 DBSSpeak operations, 36 are bidirectional, such as \( N_A \) \( \downarrow \) and \( A_N \) \( \uparrow \). The remaining three speak mode operations are apparent exceptions to the opposite pair principle. They are \( V_N \) \( \uparrow \) \( N \) \( \downarrow \) \( V \) \( s_{37} \), \( V_N \) \( \uparrow \) \( V_f \) \( s_{38} \), and \( N_V \) \( \downarrow \) \( V_i \) \( s_{39} \). As exceptions, they are only apparent because more general operations are available and used for the opposite direction. More specifically, \( V_N \) \( \uparrow \) \( N \) \( \downarrow \) \( s_{37} \) takes the more general \( N_N \) \( \uparrow \) \( V \) \( s_{24} \), \( V_N \) \( \uparrow \) \( V_f \) \( s_{38} \) the more general \( V_N \) \( \downarrow \) \( V \) \( s_{25} \), and \( V_N \) \( \downarrow \) \( V_i \) \( s_{39} \) the more general \( V_P \) \( \downarrow \) \( N \) \( s_{25} \) as their counterparts, saving three DBSSpeak operations.

### 6.6 A Remark on the Computational Complexity of DBS

An obvious requirement on the software for a talking robot is real-time performance. Of the classic complexity degrees (i) linear, (ii) polynomial, (iii) exponential, and (iv) undecidable (FoCL 8.2.1), only linear and low polynomial are computationally tractable. Therefore, it must be shown from the outset that no possible sources of high polynomial, exponential, or undecidable complexity in the worst case are lurking in the abstract software system of DBS.

Prior to adopting the agent-based ontology of DBS with the concomitant separation into the hear, think, and speak mode, LA (left-associative) Grammar (LAG) was explored as a sign-based algorithm. Functionally related to the hear mode, LAG computes possible continuations by using the total order of the incoming language surfaces (input) to establish semantic relations of structure in the output. This is in contradistinction to PSG (phrase structure) Grammar (PSG), which computes possible substitutions by always using the same abstract start symbol \( S \) as input and generating different sentences as output, whereby the derivation steps are only partially ordered.

It turned out that the complexity classes of LAG are the first, and so far the only, language hierarchy which is orthogonal to that of PSG (TCS 10.1). For example, the formal language \( a^n b^n \) is context-free (\( n^2 \) polynomial) in PSG, but \( C_1 \) (linear) in LAG. Also, the formal languages \( a^n b^n c^n \), \( a^{2^n} \), and \( a^n \) (TCS 6.1) are context-sensitive (exponential) in PSG, but \( C_1 \) (linear) in LAG.

The only source of complexity in LAG is recursive ambiguity, which occurs in such formal languages \( WW^R \) (TCS 7.2) and \( WW \) (FoCL 11.5.6). In PSG, they are in different complexity classes, i.e. context-free vs. context-sensitive, but in LAG they are in the same class \( C_2 \) (low polynomial \( n^2 \)). Other examples of recursive ambiguity in LAG are the formal languages SubsetSum (TCS 8.5, FoCL 11.5.8) and 3Sat, which are known to be inherently complex.
The unambiguous LAG for $a^k b^k c^k$ is defined as follows (CoL 8.1.6):

### 6.6.1 LA Grammar for $a^k b^k c^k$

$$LX = \{[a(a)], [b(b)], [c(c)]\}$$

$$ST_S = \{[(a) \{r_1, r_2\}]\}$$

- $r_1$: $(X) (a) \Rightarrow (aX) \{r_1, r_2\}$
- $r_2$: $(aX) (b) \Rightarrow (Xb) \{r_2, r_3\}$
- $r_3$: $(bX) (c) \Rightarrow (X) \{r_3\}$

$$ST_F = \{[\varepsilon \{r_p_3\}]\}.$$  

A lexical entry like $[a(a)]$ in the set $LX$ consists of a surface, here $a$, and a category, here $(a)$. The set $ST_S$ happens to contain only one start state, namely $\{[(a) \{r_1, r_2\}]\}$; this means that the first input must have the category $(a)$, i.e. it must have the surface $a$, and that the rules applying to the first and the second input are limited to $r_1$ and $r_2$. The rule $r_1$ adds an $(a)$ to the category, $r_2$ subtracts an $(a)$ and adds a $(b)$ to the category, while $r_3$ subtracts a $(b)$ from the category. The rule package of $r_1$ is $\{r_1, r_2\}$, i.e. after $r_1$ has applied, $r_1$ and $r_2$ are tried on the next word, and accordingly for the rules packages of $r_2$ and $r_3$. The set $ST_F$ contains only one final state, namely $\{[\varepsilon \{r_p_3\}]\}$, which means that the category must be empty ($\varepsilon$) and the currently activated rule package must be that of $r_3$.

Compared to the corresponding PS Grammar (FoCL 8.3.7), this LA Grammar is surprisingly simple. Furthermore, the LA Grammars for context-free $a^k b^k$ (CoL 10.2.3) and for context-sensitive $a^k b^k c^k$ are similar, and the number of coefficients, as in $a^k b^k c^k d^k$, $a^k b^k c^k d^k e^k$, etc., has no affect on the linear complexity of their LA Grammars.

The next word inputs to the LA Grammar for the formal language $a^k b^k c^k$ are the expressions in $LX$, e.g. $A(A)$. During parsing, the sentence start in step 4, for example, consists of the surface $A A A B$ and the category $(A A B)$:

### 6.6.2 Sample Derivation of $aaa b b b c c c$ with Active Rule Counter

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

*START=0
Expressions which are not in the language, e.g. $\text{aaabbcc}$, are analyzed to the point of the ungrammatical continuation, here $\text{aaabb+c}$, and rejected.

Applications of LA Grammar to natural language in NEWCAT, CoL, and FoCL use lexical entries such as $[\text{eat} (s3^a a^v)]$, consisting of a category and a surface, just like $[\text{a} (a)]$ in 6.6.1. Also, the LA Grammar rules for a natural language resemble those of $\text{a}^k \text{b}^k \text{c}^k$ in that they take two categories as input and produce one category and a rule package as output.

It follows from the time-linear surface compositional derivation principle of LA Grammar that the parsing of an unambiguous input consisting of n words requires n-1 successful rule applications. Rules in the subclass of C-LAGs use at most C steps in a single application, where C is a grammar-specific finite constant. Therefore, any well-formed unambiguous input to a C-LAG is parsed in linear time. The only way to increase C-LAG complexity is recursive ambiguity, which does not occur in natural language (FoCL 12.5.7).

---

7 The term time-linear refers to a derivation order, while linear time refers to a complexity degree.
The ontological foundations of LA Grammar, however, are still rooted in the sign-based approach of the linguistic and computational main stream of the time. In the agent-based approach of DBS, in contrast, the computational algorithm is integrated into a cognitive system with a database schema (memory) for storage and retrieval, a data structure for pattern matching, and external interfaces for recognition and action, all of which participate in the computational complexity of the overall system.

A factor affecting computational complexity in the hear mode is the input restrictions provided by a systematically managed now front in the agent’s memory (NLC Sect. 11.3). It allows to replace the rigid finite state network of rules and rule packages in the isolated algorithm of LA Grammar with a constantly updated, limited set of candidates for combination (1.5.1, NLC Sect. 11.3). The finite state network of rule packages (rule driven) of LAG and the limited content at the current now front (data driven) of DBS serve the same purpose; replacing the former with the latter allows to carry the results of the earlier TCS proofs for LAG over to the DBS hear mode.

A temporary factor affecting computational complexity in the speak mode is the laboratory set-up (Sect. 1.5). It is a method to separate the linguistic and computational aspects of surface production from such autonomous control issues as the what to say and the how to say it. The functional flow of the hear and the speak mode within the DBS laboratory set-up may be shown as follows (NLC 3.5.2):

**6.6.3 Functional flow within the DBS laboratory set-up**

```
<table>
<thead>
<tr>
<th>hear mode of agent B</th>
<th>speak mode of agent B</th>
</tr>
</thead>
<tbody>
<tr>
<td>surface_x</td>
<td>surface_x</td>
</tr>
<tr>
<td>interpretation</td>
<td>production</td>
</tr>
<tr>
<td>language content</td>
<td>stored in memory</td>
</tr>
</tbody>
</table>
```

The laboratory set-up stores the content derived in the hear mode in the system’s database and uses it as input to the speak mode. The surface produced by the speak mode must equal the surface analyzed in the hear mode.

Outside the laboratory set-up, an isolated algorithm for selectively activating content by navigation may be of high computational complexity. This is because more than one goal proplet may be accessible from a given proplet (multiple continuations), making the number of parallel traversals through a given content potentially exponential. However, while the hear mode is con-
trolled by the incoming data, the think mode action underlying the speak mode is controlled by the agent.

More specifically, while an agent in the hear mode must understand what is being said before being able to evaluate it, a speaker can evaluate the options of what to say and how to say it before acting on them. The latter does not require a complete consideration of each alternative provided by the content in the agent’s memory. Instead, alternatives which appear to be unattractive in the agent’s current situation may be disregarded from the start. How many options are considered and how far their consequences are being pursued is determined by the agent’s computational resources available at the moment. Whether or not the choice taken was in fact optimal is decided by the outcome.

This interaction between the agent and its environmental niche must be guided by a general principle. Instead of truth functions, as in the reasoning of symbolic logic, the overall goal of behavior in DBS is survival and reproduction (Darwin 1859). In an individual agent, it is realized as the innate drive to maintain a state of balance vis à vis a constantly changing external and internal environment. The purpose of maintaining or regaining balance includes such human desires as for acceptance, love, belonging, freedom, and fun, which may be subsumed under the balance principle by treating them as part of the agent’s internal environment, like hunger.

When the DBS laboratory set-up is replaced by the balance principle, it is the agent’s general behavior control which provides the speak mode with the what to say and how to say it:

6.6.4 Functional flow outside the DBS laboratory set-up

To ensure a well-functioning combination of language and non-language cognition, they are designed computationally to use the same data structure, the same database schema, the same time-linear algorithm based on the same method of matching between the pattern and the content level, and the same recognition and action concepts (grounding).
List of Examples

The following test list is intended as a quick overview. For a more detailed linguistic description of the examples and the systematic reasons behind their choice and ordering see Sect. 1.6.

2.1.1 I slept.
2.2.1 The old dog had been sleeping.
2.3.1 I saw you.
2.4.1 The big dog saw a small bird.
2.5.1 That Fido barked amused Mary.
2.6.1 Mary heard that Fido barked.

3.1.1 Perhaps Fido is still sleeping there now.
3.2.2 Fido ate the bone on the table (adverbial)
3.3.1 The dog which saw Mary barked.
3.4.1 The dog which Mary saw barked.
3.5.1 When Fido barked Mary laughed.
3.6.1 Fido, Tucker, and Buster snored loudly.

4.1.1 Julia put the flowers on the table.
4.2.1 The letter made Mary happy.
4.3.1 Fido dug the bone up.
4.4.1 John tried to read a book.
4.5.1 Julia is a doctor.
4.6.1 Fido is hungry.

5.1.1 Fido ate the bone on the table under the tree in the garden...
5.2.1 Bob bought an apple, # peeled a pear, ..., and # ate a peach.
5.3.1 Bob bought an apple, Jim # a pear, ..., and Bill # a peach.
5.4.1 Bob bought #, Jim peeled #, ..., and Bill ate a peach.
5.5.1 Whom does John say that Bill believes ... that Mary loves?
5.6.1 Mary saw the man who loves the woman ... who fed Fido.

---

8 Online, the preceding three-digit numbers allow comfortable access via automatic cross-referencing.
References


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


Brömel, A. v. (1794) Wie machen sie’s in der Komödie? Wien: Wallishauser


Cooper, R. (1983) Quantification and Syntactic Theory, Springer


For more extensive bibliographies on related topics see FoCL (20 pages), NLC (15 pages), and CLaTR (20 pages).
Name Index

Bachman, C.W., 5
Bar-Hillel, Y., 162
Barsalou, L., 154
Brömel, A. v., 182

Chomsky, N., 191
Cooper, R., 236

Elmasri, R., 4
Everett, D., 191

Fitsch, W., 191
Hauser, M., 191

Levin, B., 14
Moffat, A., 5
Navathe, S.B., 4

Zobel, J., 5
Subject Index

active, 11
ambiguity, 254
arc numbering, 14
breadth first traversal, 16
clausal adnominal, 107, 116
clausal adverbial, 126
clausal coordination, 17
clausal functor-argument, 17
clausal modification, 13
clausal object, 77
clausal relation, 10
clausal subject, 65, 132
connectives of hear mode, 4
connectives of speak mode, 7
content, 11, 12, 33, 39, 48, 54, 65, 75, 85, 97, 107, 116, 126, 135, 147, 156, 164, 171, 180, 186, 193, 204, 221, 237, 252, 255, 272
content level, 4
content-addressable, 5
Continuity Condition, 8, 14, 94, 207
database schema, 5
depth first traversal, 15
elementary relation, 10
extrapropositional coordination, 13
gapping, 204
grammatical complexity, 10
grounding, 156
hear mode #-marking, 105
inverted file, 5

LA Grammar
\(a^k b^k c^k\), 313
laboratory set-up, 19
lexical proplets, 1
loom-like clearance, 6, 30, 289
mental state verbs, 75
mood adnominals, 156
multiple elementary adverbials, 85
NAG, 8
now front clearance, 6
object gapping, 18, 245
operations sequence, 51
paraphrase, 10, 12
passive, 11
pattern level, 4
phrasal modification, 97
phrasal object, 58
phrasal relation, 10
phrasal subject, 54
phrasal verb form, 39
predicate gapping, 222, 230
prepositional phrase, 153, 157, 158, 160
psych verbs, 65
repeating phrasal modifiers, 21
sediment, 6
semantic relations of structure, 7
signature, 8
speak mode #-marking, 105
SRG, 8, 10
subject gapping, 17, 19, 214
subject/verb agreement, 43
surface realization, 8, 51
to-do list, 20
traversal direction, 115
traversal order, 14
unbounded dependency, 21, 262
variable markings, 31
word bank, 5
word order difference, 13
word order variation, 10