Foundations of Lexical Resource Semantics

Frank Richter Seminar für Sprachwissenschaft Abteilung Computerlinguistik Eberhard-Karls Universität Tübingen

Habilitationsschrift

2004

Contents

In	trodu	uction		1				
Ι	The Mathematical and the Linguistic Framework Model-Theoretic Grammar							
1	The Mathematical Framework: RSRL							
	1.1	Descri	ption	15				
	1.2	Meanin	ng	47				
	1.3	RSRL	without Relational Monsters	51				
2	The Linguistic Framework: HPSG							
	2.1	Struct	ural Hypotheses in Syntax	62				
		2.1.1	The Standard Hypotheses	63				
		2.1.2	Unembedded Signs	74				
		2.1.3	Summary: A Grammar and an Intended Model	78				
	2.2	Toward	d Normal Form Grammars	82				
		2.2.1	Three Types of Problems	84				
		2.2.2	Solutions	90				
		2.2.3	A Symbolization of the Results	105				
	2.3 A Normal Form Grammar with Exhaustive Model							
	2.4 Alternative Notions of Meaning							
		2.4.1	Linguistic Types as Abstract Feature Structures	118				
		2.4.2	Canonical Representatives as Singly Generated Models	123				
		2.4.3	The Third Way: Unique Unembedded Sign Configurations	128				
3	Technicalities							
	3.1	Notati	onal Conventions	137				
	3.2	Equiva	lence Proof	142				
тт	S.	omont	tics	151				
ΤŢ	Do Mod	ol Theo	LICO	152				
	mou	CI-T HEC		100				

4	\mathbf{Log}	ical Languages as Object Languages of RSRL	163			
	4.1	Ty2 in RSRL	164			
	4.2	Ty2 in Normal Form HPSG Grammars	179			
5	Lexicalized Flexible Ty2					
	5.1	Semantic Composition	188			
	5.2	Lexicalized Flexible Ty2	196			
	5.3	Discussion	206			
6	Lexical Resource Semantics					
	6.1	Empirical Motivation	214			
	6.2	The Architecture	217			
	6.3	Outlook	229			
Bibliography						

Chapter 1

The Mathematical Framework: RSRL

In this chapter I will introduce a new dialect of Relational Speciate Re-entrant Language (RSRL). RSRL provides a mathematical framework for writing fully explicit HPSG grammars with a model-theoretic interpretation. Having a mathematical framework of this nature is a necessary prerequisite for the integration in HPSG grammars of standard semantic representation languages such as the languages of Two-sorted Type Theory (Ty2), which we will define as object languages of RSRL theories.

RSRL is a formalism which was built for a specific purpose. Its raison d'être is HPSG. Despite this historical fact, RSRL does not comprise any genuinely linguistic assumptions. It was not conceived as a linguistic framework. The only conceptual restrictions which come with the formalism concern certain very abstract epistemological assumptions about its logical interpretations. While these assumptions make the formalism more or less appropriate for analyzing natural languages depending on the scientist's epistemological premises about languages, these restrictions may be generalized for the application of the RSRL formalism to any empirical domain about which scientists might make diverging basic assumptions of a similar kind. RSRL could be used for theories from any empirical domain which can reasonably be approached under assumptions which are compatible with the techniques of classical logical description, provided its logical languages are expressive enough for the task. According to the scientific meta-theory adopted by researchers in the HPSG community, linguistic hypotheses are expressed not in the mathematical framework, but in the linguistic framework. With the exception of a certain number of meta-assumptions, the linguistic theory of HPSG is expressed as a logical theory in a given mathematical framework. Chapter 2 will offer a survey of some of the fundamental linguistic tenets of HPSG, and there I will also discuss substantive linguistic hypotheses. We will then see how the linguistic framework delimits the space of possible HPSG grammars compared with the much larger space of possible RSRL grammars.

In contrast to earlier expositions of RSRL, the present chapter defines its class of languages in terms of the syntax of attribute value matrices (AVMs). This is the syntax which is generally adopted in the linguistic HPSG literature. Previous presentations of RSRL usually adopted logical languages which were an extension of Speciate Re-entrant Logic (SRL, [King, 1989, King, 1999]), the precursor of RSRL, and proceeded to a definition of an AVM syntax which was interpreted by a translation into the original syntax. The textbook [Richter, 2004b], which is an introduction to the mathematical framework of HPSG and to grammar writing and grammar implementation, uses a syntactic dialect of RSRL which is an extension of the description language provided by the grammar implementation platform TRALE in order to make the connection between theoretical grammar development and computational applications more transparent. While the RSRL dialect of the textbook is motivated by bridging the gap between different areas of HPSG research, and whereas the predominantly mathematical orientation of the earlier expositions favored an indirect approach with an extension of the SRL syntax as the primary logical languages, keeping an SRL-based syntax as an intermediate layer between AVM syntax and interpretation would have several drawbacks in the present context.

First of all, it is not practically feasible to write realistic grammatical principles in the original syntax of RSRL for an audience of linguists. The linear syntactic notation of RSRL is much more opaque than the two-dimensional AVM syntax common in linguistics when descriptions get as large as they usually do in linguistic constraints. The conventional AVM syntax makes the logical structure of the statements much more transparent by collapsing most conjunctions into two-dimensional matrices, which are easily perceived as units. Moreover, most readers are very familiar with the AVM syntax whereas they are usually completely unfamiliar with the syntax of the original logical languages of the more mathematically oriented literature.

This situation virtually necessitates the introduction of an AVM syntax in addition to the RSRL syntax in presentations of grammars intended for a linguistic audience. But to follow the well trodden path of putting an additional uninterpreted syntactic layer on top of the logical syntax entails extra work in setting up the technical machinery. In addition to the interpreted formal languages, we need to define the AVM description languages for the linguistic users as well as a purely syntactic rewriting system for the expressions of the user languages. For the readers of grammars written in the user languages this has the very unpleasant consequence that it becomes harder to understand what the AVM descriptions really mean in terms of the semantics given to the constraint languages. The reader is forced to compute a fairly complex translation of the familiar AVM syntax into an unfamiliar syntax before arriving at the interpretation of the constraints. Understanding the meaning of RSRL descriptions is, however, fundamental to a proper understanding of the proposals put forth in this work concerning the construction of an adequate architecture of semantic representations in constraint-based grammar. For these reasons it seems a very worthwhile enterprise to define a new dialect of RSRL which directly interprets a syntax of AVM expressions without the detour through a syntactic translation. Proceeding in this way is made even more attractive by the fact that it is possible to give an AVM syntax an interpretation which matches the pre-theoretical understanding of these expressions very well. A direct semantics of the AVM expressions simply needs to provide precise definitions which are supported by an already existing informal understanding of their meaning. As a result, for those readers who are interested in subtle technical details, it will become much

easier to check the exact meaning of grammatical principles.

The syntax which will be defined in the coming section will closely follow the syntax of AVM formulae introduced in [Richter, 2004a, pp. 212–214], but will omit special constructs for lists and chains.¹ Since the syntax of our formal languages will be different from the original RSRL syntax, it cannot be taken for granted that we will actually be defining a dialect of RSRL. The languages which we will define could inadvertently be either more or less expressive than the original languages. However, we will be able to show that this is not the case. Because of the close relationship between the present syntax and the uninterpreted syntax of AVM formulae from [Richter, 2004a], the translation function from AVM expressions to RSRL expressions, which is employed there to give meaning to AVM expressions indirectly by translating them to the interpreted language, can be used to show that the new syntax and semantics indeed define a variant of RSRL. Section 3.2 sketches a proof of this claim. This proof ultimately justifies my terminology in the coming sections, in which I will treat the present formalism as an RSRL formalism.

In the following exposition of RSRL I will refrain from presenting a motivation of its architecture and constructs. This has already been done thoroughly in [Richter, 2004a]. The reason for creating the RSRL formalism was to provide mathematical foundations for the actual practice of grammar specifications found in the HPSG literature. The goal was to interpret the linguistic specifications as literally as possible in the way in which linguists write them down and explain their meaning. In particular, it was never the purpose of RSRL to find the minimally necessary technical apparatus to write HPSGinspired grammars. As a consequence of this approach, the languages of RSRL are very expressive and flexible, which turns out to be a considerable advantage for developing new modules of grammar.

The direct interpretation of the *lingua franca* of AVM expressions in the following new RSRL dialect should make subsequent chapters equally accessible to readers who are interested in the mathematical foundations of LRS, and to readers who are more casually interested in the technical details but wish to focus on the linguistic content of the theory. Although the text will not provide a motivation for the architecture of the formalism, readers who are unfamiliar with its previous incarnation but fluent in current work on HPSG will find enough illustrations of each syntactic and semantic construct to see how the interpretation of linguistic principles proceeds. Combining the definitions with many concrete linguistic examples, the text will provide a self-contained exposition of the formalism in which we will ultimately specify LRS.

1.1 Description

The most important property which distinguishes RSRL from familiar logical languages such as first order predicate logic is the meaning of its expressions. In first order predicate logic, expressions denote the truth values True or False. In RSRL expressions denote sets of objects. It belongs to the family of description logics.

¹They are added as syntactic sugar for the convenience of grammar writers in Section 3.1.

The second important fact to note about RSRL involves the kinds of structures for whose characterization its languages are intended. A crucial motivation for designing a formalism for HPSG grammars of the kind of RSRL comes from the way in which the entities which it is supposed to characterize are conceptualized. There are important intuitions about the structure and properties of natural language expressions which underly the architecture of this particular description logic. Since our ultimate domain of application will be linguistics, I will draw all of my examples from linguistics. The reader should, however, bear in mind that the choice of the application domain is left to the user and in no way determined or even suggested by the formalism.

The central idea is that the universe of (linguistic) entities is populated by collections of *connected configurations*, or *connected structures*. For the sake of concreteness we may picture these connected structures as collections of entities which are connected by arcs. For each connected structure there exists one entity from which we can reach all others by following a sequence of arcs. For the sake of convenient terminology, I will occasionally refer to this entity as the *topmost* entity. In a pre-theoretical sense, entities which can be reached from some other entity e by following sequences of arcs are 'components', or 'constituents', of e. It is not so much single entities which are of interest, but rather these connected structures consisting of collections of entities which are held together by arcs. The meaning ascribed to arcs by linguists roughly corresponds to 'is a property of', 'belongs to' or 'is a constituent of'. The preferred interpretation might actually vary for different attributes and attribute values.

The most prominent connected structures of interest in linguistics are signs (although there might be others). The topmost entity in a collection of entities which make up a sign is typically a phrase. Phrases contain, of course, other phrases and ultimately words amongst their components. Each one of the phrases and words consists in turn of other entities such as syntactic category entities, case entities, verb form entities, semantic representations, subcategorization lists, slash sets, and more. Whereas every entity in a connected structure is accessible from the topmost phrase which it contains, it is not necessarily the case that the phrase can be reached from any of its 'components'.

Example (1) shows a conceivable connected configuration of the kind we have described. The topmost entity in this concrete configuration is the leftmost node in the graph. It is labeled *word*. The other nodes in the graph are the other entities in the connected configuration, and they are all accessible from the topmost entity by following a sequence of arcs. Since the topmost entity in this configuration is a *word* entity, and since words, as well as phrases, count as instances of signs, the entire configuration is considered a sign. The component nodes of the word are interpreted as its properties. Which property each constituent signifies is determined by its label and by the labels of the arcs which lead to it from the topmost entity in the configuration.



The linguistic universe is, of course, typically populated by a great number of or infinitely many connected configurations.

Three significantly different model theories for RSRL grammars have been proposed to date [King, 1999, Pollard, 1999, Richter, 2004a]. The major differences between them concern the philosophical assumptions about the ontological status of the intended models of grammars. These assumptions ultimately affect how the relationship between the intended models of grammars and the domain of empirical phenomena is to be characterized. They also determine the ontological status of the entities in connected configurations such as the one depicted in (1), and the status and nature of the connected configurations as a whole. I will postpone a discussion of various aspects of these foundational issues to Section 1.2 and Section 2.4, when we will have seen enough of the mathematical and linguistic frameworks to discuss them in a meaningful way. Inspite of what we will then have to say about the underlying ontological assumptions, there is one theory of the meaning of grammars which makes the intuitions about connected structures discussed above particularly clear at first sight. This theory postulates that so-called *abstract feature structures* are the objects in the denotation of grammars, thus capturing the assumptions about the connectedness of the intended structure in a particularly salient manner. The objects of interest are (equivalence classes of) a particular kind of connected graph. For reasons explained in Section 2.4 I will not use abstract feature structures as modeling structures, although the intuitions about the connectedness of the elements in the modeling domain remain the same in all relevant respects. To understand the intentions of HPSG linguists it is quite enlightening to realize straight away the fact that possible relationships between different connected structures in the universe are *a priori* of no interest to them. The formalism will not even provide any means to talk about relationships of entities in the modeling domain across the boundaries of connected structures. Due to the architecture of the relevant interpreting structures envisioned by linguists, the formalism is tailored for talking about collections of separate connected structures in a universe of entities.

Let me briefly sketch the basic architecture of the RSRL formalism before starting with the definitions. Just as in any standard logical formalism, RSRL provides a class of formal languages. Each language is based on a signature which provides the nonlogical symbols for the formulae. The nonlogical symbols of the signature generate a set of formulae. A grammar is a pair consisting of a signature and a set of descriptions. In order to mirror the structure of AVM expressions in our formulae, the syntactic structure of our expressions will be defined in two steps. The inner layer of our syntax will consist of the familiar matrices of HPSG. In an outer layer of syntax we will add relational expressions, quantification and the possibility to combine matrices with the standard logical connectives. The interpretation of the formulae will mirror the two syntactic layers. The satisfaction relation of complete AVM formulae will be based on the satisfaction relation of the matrices which are part of them. In a first approximation of the meaning of grammars we will finally define the notion of a model of a grammar. Essentially, an interpretation satisfies each description in the grammar. In Section 1.2 we will discuss why models are only a first step toward the meaning of grammars, and we will supply a more elaborate notion of the meaning of grammars.

In our description languages we will want to use variables. For this purpose we give ourselves a countably infinite supply of variables:

Definition 1 \mathcal{VAR} is a countably infinite set of symbols.

I will call each element of \mathcal{VAR} a variable. In the usual linguistic notation of AVM expressions, variables are usually written as integers in boxes such as [2], [5], and [27]. Following the linguistic conventions, I will sometimes call these variables in the form of boxed integers tags. I will assume that the symbol ':' is a reserved symbol and not a variable. I will later use this reserved symbol in a way similar to variables. Therefore I will adopt the convention of referring to the union of a set of variables, \mathcal{VAR} , and the singleton set containing ':' by $\mathcal{VAR}^{:}$.

Before we can define the syntax of expressions we need the relevant sets of nonlogical constants. In HPSG these are sort symbols, attribute symbols and relations symbols of an arbitrary arity. In addition the sorts are arranged in a sort hierarchy, and sorts and attributes are related to each other by in turn declaring certain attributes appropriate for certain sorts and giving pairs of sorts and appropriate attributes appropriate sort values. I will obtain these additional structures by introducing a partial order over the sort symbols and by providing a partial function called appropriateness function, which is subject to certain conditions which enforce the usual attribute and attribute value inheritance of the feature declarations of HPSG grammars. Signatures provide the necessary sets of symbols together with the additional relationships:

¹ Definition 2 Σ is a signature iff

²
$$\Sigma$$
 is a septuple $\langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$,

 $\langle \mathcal{G}, \sqsubseteq \rangle$ is a partial order,

$$\mathcal{S} = \left\{ \sigma \in \mathcal{G} \middle| \begin{array}{l} \textit{for each } \sigma' \in \mathcal{G}, \\ \textit{if } \sigma' \sqsubseteq \sigma \textit{ then } \sigma = \sigma' \end{array} \right\}$$

 $5 \qquad \mathcal{A} \text{ is a set,}$

4

- \mathcal{F} is a partial function from the Cartesian product of \mathcal{G} and \mathcal{A} to \mathcal{G} ,
- for each $\sigma_1 \in \mathcal{G}$, for each $\sigma_2 \in \mathcal{G}$ and for each $\alpha \in \mathcal{A}$,
- 8 9

if $\mathcal{F}\langle \sigma_1, \alpha \rangle$ is defined and $\sigma_2 \sqsubseteq \sigma_1$ then $\mathcal{F}\langle \sigma_2, \alpha \rangle$ is defined and $\mathcal{F}\langle \sigma_2, \alpha \rangle \sqsubset \mathcal{F}\langle \sigma_1, \alpha \rangle$,

10 \mathcal{R} is a finite set, and

$$\mathcal{AR}$$
 is a total function from \mathcal{R} to \mathbb{N}

I call each element of \mathcal{G} a sort, $\langle \mathcal{G}, \sqsubseteq \rangle$ the sort hierarchy, each element of \mathcal{S} a maximally specific sort or a species, each element of \mathcal{A} an attribute, \mathcal{F} the appropriateness function, each element of \mathcal{R} a relation symbol, and $\mathcal{A}\mathcal{R}$ the arity function. I say that sort σ_2 is at least as specific as sort σ_1 iff $\sigma_2 \sqsubseteq \sigma_1$; conversely, I call σ_1 a supersort of σ_2 and say that σ_1 subsumes σ_2 . The symbols $\forall, \exists, :, \neg, [,], \land, \lor, \rightarrow, \leftrightarrow, \dagger, \triangleright,$ chain, echain, nechain and metatop are reserved symbols, and I assume that none of them are a variable, a sort, an attribute or a relation symbol. \mathbb{N} is the set of positive integers. I do not allow relations of arity zero. The appropriateness function \mathcal{F} respects the usual restrictions, which go by the name of attribute inheritance in linguistics: If an attribute α is appropriate to a sort σ_1 (" $\mathcal{F}\langle\sigma_1, \alpha\rangle$ is defined", line 8), it is also appropriate to all sorts σ_2 subsumed by σ_1 (" $\mathcal{F}\langle\sigma_2, \alpha\rangle$ is defined", line 9), and the value that the appropriateness function yields for α at σ_2 is at least as specific as the value for α at σ_1 (" $\mathcal{F}\langle\sigma_2, \alpha\rangle \sqsubseteq \mathcal{F}\langle\sigma_1, \alpha\rangle$ ", line 9).

By convention I write attributes in SMALL CAPITALS, sorts in *italics*, and relation symbols in typewriter font. I can thus distinguish between the attribute SYMBOL, the sort *symbol* and the relation symbol without having to say explicitly which is which.

All of this can best be seen in a small example, which I will take from linguistics. We will consider a signature Σ_1 that will be sufficient for writing a very simple grammar predicting the phrase structural composition of the sentence *Uther walks*. The grammar will include a lexicon of the two words, an indication of their subcategorization properties, and it will mark the syntactic head of the single sentence captured by the grammar. The grammar that we can build based on this simple signature will actually be sufficiently complex to form a reasonable basis for a discussion of the model theory of HPSG grammars in the RSRL formalism in Chapter 2.

Now let Σ_1 be $\langle \mathcal{G}_1, \sqsubseteq_1, \mathcal{S}_1, \mathcal{A}_1, \mathcal{F}_1, \mathcal{R}_1, \mathcal{AR}_1 \rangle$ with

$$\mathcal{G}_{1} = \left\{ \begin{array}{l} top, sign, phrase, word, list, nelist, elist, cat, \\ head, verb, noun, phonstring, uther, walks \end{array} \right\}, \\ \left\{ \begin{array}{l} \langle top, top \rangle, \langle sign, sign \rangle, \langle phrase, phrase \rangle, \langle word, word \rangle, \langle list, list \rangle, \\ \langle nelist, nelist \rangle, \langle elist, elist \rangle, \langle cat, cat \rangle, \langle head, head \rangle, \langle verb, verb \rangle, \\ \langle noun, noun \rangle, \langle phonstring, phonstring \rangle, \langle uther, uther \rangle, \langle walks, walks \rangle, \\ \langle sign, top \rangle, \langle phrase, top \rangle, \langle word, top \rangle, \langle list, top \rangle, \langle nelist, top \rangle, \\ \langle elist, top \rangle, \langle cat, top \rangle, \langle head, top \rangle, \langle verb, top \rangle, \langle noun, top \rangle, \\ \langle phonstring, top \rangle, \langle uther, top \rangle, \langle walks, top \rangle, \langle phrase, sign \rangle, \langle word, sign \rangle, \\ \langle walks, phonstring \rangle \end{array} \right\},$$

$$\begin{split} \mathcal{S}_{1} &= \{phrase, word, nelist, elist, cat, verb, noun, uther, walks\}, \\ \mathcal{A}_{1} &= \{PHON, H_DTR, NH_DTR, FIRST, REST, HEAD, SUBCAT\}, \\ \mathcal{A}_{1} &= \left\{ \begin{array}{l} \langle \langle sign, PHON \rangle, list \rangle, \langle \langle phrase, PHON \rangle, list \rangle, \langle \langle word, PHON \rangle, list \rangle, \\ \langle \langle phrase, H_DTR \rangle, sign \rangle, \langle \langle phrase, NH_DTR \rangle, sign \rangle, \langle \langle sign, CAT \rangle, cat \rangle, \\ \langle \langle phrase, CAT \rangle, cat \rangle, \langle \langle word, CAT \rangle, cat \rangle, \langle \langle nelist, FIRST \rangle, list \rangle, \\ \langle \langle nelist, REST \rangle, list \rangle, \langle \langle cat, HEAD \rangle, head \rangle, \langle \langle cat, SUBCAT \rangle, list \rangle \end{array} \right\}, \\ \mathcal{R}_{1} &= \{ \text{append, member} \}, \text{ and } \\ \mathcal{A}\mathcal{R}_{1} &= \{ \langle \text{append, } 3 \rangle, \langle \text{member}, 2 \rangle \}. \end{split}$$

Making the set of sorts, S, an explicit part of the signature is obviously redundant, since we could always recover S as a subset of the set of all sorts, G, by investigating the sort hierarchy, $\langle G, \sqsubseteq \rangle$. Taking a look at our example will illustrate this: we can find the elements of S_1 by collecting all the elements of the pairs in \sqsubseteq_1 which occur only on the left hand side in the pairs, except when they occur left *and* right. These are obviously the most specific elements in the partial order. In later definitions, however, we will often have occasion to refer to the set of species, and it is convenient to have a designated symbol for it.

It is also obvious that stating a signature in the format given above makes the signature hard to read. For easier readability I will therefore adopt a graphical notation, which is a variant of the notation introduced in the *Morph Moulder* (MoMo) system [Richter et al., 2002, Richter, 2004b].² According to the variant of the notational conventions of MoMo which I will adopt throughout this work, the signature Σ_1 is notated as shown in Figure 1.1.

The subsumption relation between sorts is indicated by indentation. Appropriate attributes are listed in a column behind the highest sort σ_1 in the hierarchy for which they are appropriate, together with the sort σ_2 appropriate for σ_1 at that attribute. The inheritance of these attributes and attribute values at lower sorts in the hierarchy is left implicit in order to keep the notation compact. The notation in Figure 1.1 is thus to be understood as saying that PHON and CAT are not only appropriate to *sign* but are also appropriate to *phrase* and *word*. Moreover, the appropriate values remain *list* and *cat* respectively, since there is no more specific value specified for this attribute behind the subsorts *phrase* and *word*. Relation symbols are listed at the end of the specification, together with the arity assigned to them by the arity function.

For the following definitions it will be convenient to employ a small number of notational conventions in order to keep the notation compact. CONVENTION 1 introduces four notations to refer to sets which we will frequently need:

Convention 1 Suppose that S is a set. I will observe the following notations:

1. For each $n \in \mathbb{N}$, S^n is the n-fold Cartesian product $S_1 \times \ldots \times S_n$.

 $^{^{2}}$ MoMo is a teaching software for studying RSRL by writing RSRL grammars and constructing interpretations for descriptions.

```
top
                 list
   sign
         PHON
          CAT
                  cat
      phrase
               H_DTR
                         sign
               NH_DTR
                         sign
      word
   list
      nelist
              FIRST
                      top
              REST
                      list
      elist
   cat
        HEAD
                   head
         SUBCAT
                  list
   head
      verb
      noun
   phonstring
      uther
      walks
Relations
member/2
append/3
```



- 2. S^* is the set of finite sequences of elements of S. The reader may also look at S^* as the union of the n-fold Cartesian products over S, including the empty product.
- 3. S^+ is the set of nonempty finite sequences of elements of S.
- 4. \overline{S} is shorthand for the set $S \cup S^*$.

With these conventions in hand we can now interpret signatures. The interpretations of signatures provide sets of entities from the universe. They interpret sort symbols by labeling all entities from the universe with species, they take attributes to be partial functions on the entities from the universe, and they interpret n-ary relation symbols as sets of n-tuples of entities and sequences of entities:

Definition 3 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, I is a Σ interpretation iff

- I is a quadruple $\langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,
- U is a set,
- S is a total function from U to $\mathcal{S},$
- A is a total function from \mathcal{A} to the set of partial functions from U to U,

for each $\alpha \in \mathcal{A}$ and each $u \in \mathsf{U}$,

if
$$A(\alpha)(u)$$
 is defined
then $\mathcal{F}(S(u), \alpha)$ is defined, and $S(A(\alpha)(u)) \sqsubseteq \mathcal{F}(S(u), \alpha)$

for each $\alpha \in \mathcal{A}$ and each $u \in \mathsf{U}$,

if
$$\mathcal{F}(\mathsf{S}(u), \alpha)$$
 is defined then $\mathsf{A}(\alpha)(u)$ is defined, and

R is a total function from \mathcal{R} to the power set of $\bigcup_{n \in \mathbb{N}} \overline{U}^n$, and

for each
$$\rho \in \mathcal{R}$$
, $\mathsf{R}(\rho) \subseteq \overline{\mathsf{U}}^{\mathcal{AR}(\rho)}$.

U is the set of entities in the universe. I call S the species assignment function, A the attribute interpretation function, and R the relation interpretation function. According to DEFINITION 3, a Σ interpretation consists of a set of entities which are all labeled by species. The attribute interpretation function respects the demands of the appropriateness function: If the interpretation of an attribute α is defined on an entity u, then the attribute α must be appropriate to the species of u; and the species of the entity u' that we obtain by interpreting α at u must be a subsort of the sort which is appropriate for α at the species of u. Conversely, if an attribute α is appropriate to the species of a given entity, then the interpretation denoted by an attribute is defined for all or no entities of any given species. To simplify my terminology, I will usually call the partial function designated by an attribute α the function α . Similarly, if S labels an entity u with species σ by mapping it to σ , I will simply say that entity u has species σ .

R interprets each relation symbol as a set of tuples of the form $\langle u_1, \ldots, u_n \rangle$, where: (a) each u_i is an entity in U, or (b) each u_i is a sequence of entities in U, or (c) some u_i are entities and others are sequences of entities in U. The number n is determined by the arity of the relation as given by the arity function. As witnessed by the cases (b) and (c), arguments of relations may be sequences of entities in U. I will call these sequences of entities *chains* (of entities).

The original motivation for the introduction of chains to the formalism was the fact that linguists tend to make one exception to the rule that they only describe entities connected by attributes: In arguments of relations such as the **shuffle** relation of linearization grammars, and also in other relations involving lists and sets, linguists sometimes refer to apparent lists or sets which actually do not occur in the described connected entities, although each of the individual elements does. It is for the purpose of capturing these cases that chains were introduced into HPSG, and are thus allowed in the arguments of relations. For a deeper elaboration and a thorough discussion of chains in HPSG grammars, see [Richter, 2004a, Section 4.3].

 Σ interpretations are best pictured as graphs in which the nodes are the entities of the universe, the sort labels are written next to the nodes, and the partial functions named

by attributes are pictured as arrows labeled by their attribute names. The origin of each arrow is an element in the domain of the function named by the attribute, and the node it points to is the element in the range of the function to which it maps the entity at its origin. Picturing relations is a bit more difficult, because there is no good way of marking the tuples of elements in relations in a graph without making it unreadable. For this reason, whenever I want to indicate in a Σ interpretation which nodes are in a relation, I will assign numbers to the nodes and state the relation by stating the tuples of numbers which I assigned to the nodes in the relation.

The two examples of Σ_1 interpretations in (2) below illustrate these conventions. The first example, (2a), presents a Σ_1 interpretation with empty relations member and append. Since there are no elements in the relations in (2a), it is not necessary to label the nodes in the first example with numbers. The second example, (2b), illustrates a case in which there are tuples in the member and in the append relation, and the nodes are supplied with numbers, which may then be referred to for the specification of the relations. Note that the elements in the member relation do not correspond at all to an intuitive understanding of what should be in a member relation if member is a relation between the elements of lists and the lists that these elements are on. Neither is the single triple in the append relation intuitively sensible. As we will see later, the meaning of relation symbols is determined by the theory of these symbols in grammars; it is not fixed in advance for arbitrary interpretations of a signature. The reader might also notice that the Σ_1 interpretation of (2b) contains some more potentially surprising properties: while the configuration of connected entities under the entity 0, which is labeled *word*, can easily be recognized as what linguists would think of as the word *Uther*, the orphaned entity 7 with sort label *walks* and the entity 8 with sort label *cat* and the small connected structure beneath it might initially not be expected to appear in interpretations of signature Σ_1 . It should furthermore be noted that two SUBCAT arrows point to the *elist* entity 6, meaning that the partial function named by the attribute symbol SUBCAT maps the *cat* entity 8 and the *cat* entity 4 on the *elist* entity 6. We will have occasion to pursue these observations further when we discuss the meaning of grammars in Section 2.2 below.

(2) a. A Σ_1 interpretation with empty relations:



b. A Σ_1 interpretation with nonempty relations:



 $\mathtt{append} = \{ \langle 8, 8, 8 \rangle \}$

Before I can start to define the syntax of the logical languages generated by signatures, I need to augment the signatures by symbols for reasoning about *chains*. For this purpose I will use the reserved symbols *chain*, *echain*, *nechain* and *metatop*, which will be added to the repository of sort symbols, and the reserved symbols \dagger and \triangleright , which will be added to the stock of attribute symbols. The former are quasi-sorts, and the latter are quasi-attributes.

The quasi-sorts *chain*, *echain* and *nechain* form a small chain hierarchy, with *echain* and *nechain* below *chain*. The symbol *metatop* is used as a top element of the extended sort hierarchy, where for any signature Σ with sort hierarchy $\langle \mathcal{G}, \sqsubseteq \rangle$, all elements of \mathcal{G} are below *metatop*, just as *chain*, *echain* and *nechain*. The quasi-attributes \dagger and \triangleright will be used to describe chains and to refer to elements from chains. They correspond to the familiar attributes FIRST and REST of HPSG grammars, which are used in the description of lists.

DEFINITION 4 provides these augmentations for any signature:

Definition 4 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$,

$$\begin{split} \widehat{\mathcal{G}} &= \mathcal{G} \cup \{ chain, echain, nechain, metatop \}, \\ \widehat{\square} &= \square \cup \{ \langle echain, chain \rangle, \langle nechain, chain \rangle \} \cup \{ \langle \sigma, \sigma \rangle \mid \sigma \in \widehat{\mathcal{G}} \backslash \mathcal{G} \} \\ &\cup \{ \langle \sigma, metatop \rangle \mid \sigma \in \widehat{\mathcal{G}} \}, \\ \widehat{\mathcal{S}} &= \mathcal{S} \cup \{ echain, nechain \}, and \\ \widehat{\mathcal{A}} &= \mathcal{A} \cup \{ \dagger, \triangleright \}. \end{split}$$

Note that if $\langle \mathcal{G}, \sqsubseteq \rangle$ is a finite partial order, then $\langle \hat{\mathcal{G}}, \widehat{\sqsubseteq} \rangle$ is also a finite partial order. This is important since the sort hierarchies of actual HPSG grammars are finite partial orders with a top element. I call $\hat{\mathcal{G}}$ the expanded sort set, $\langle \hat{\mathcal{G}}, \widehat{\sqsubseteq} \rangle$ the expanded sort hierarchy, $\widehat{\sqsubseteq}$ the expanded sort hierarchy relation, $\hat{\mathcal{S}}$ the expanded species set, and $\hat{\mathcal{A}}$ the expanded attribute set.

It is now necessary to explain how we will interpret the new symbols of the chain hierarchy and the quasi-attributes. According to DEFINITION 5, the new maximally specific quasi-sorts *echain* and *nechain* will be used to label the empty sequence of entities and any nonempty sequence of entities in the universe. The quasi-attribute \dagger is interpreted as the function which maps a sequence of entities to its first element, and \triangleright is interpreted as the function which maps a sequence of entities to another sequence of entities which is just like the first one, except that its first element is cut off.

The interpretation of the other new symbols, *chain* and *metatop* will follow from the interpretation of (quasi-) sort symbols which are not maximally specific in formulae.

Definition 5 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, for each Σ interpretation $I = \langle U, S, A, R \rangle$,

 $\widehat{\mathsf{S}}$ is the total function from $\overline{\mathsf{U}}$ to $\widehat{\mathcal{S}}$ such that

for each $u \in U$, $\widehat{S}(u) = S(u)$, for each $u_1 \in U$, ..., for each $u_n \in U$, $\widehat{S}(\langle u_1, \dots, u_n \rangle) = \begin{cases} echain & \text{if } n = 0, \\ nechain & \text{if } n > 0 \end{cases}$, and $\widehat{\mathsf{A}}$ is the total function from $\widehat{\mathcal{A}}$ to the set of partial functions from $\overline{\mathsf{U}}$ to $\overline{\mathsf{U}}$ such that

for each
$$\alpha \in \mathcal{A}$$
, $\widehat{\mathsf{A}}(\alpha) = \mathsf{A}(\alpha)$,
 $\widehat{\mathsf{A}}(\dagger)$ is the total function from U^+ to U such that for each
 $\langle u_0, \dots, u_n \rangle \in \mathsf{U}^+$,
 $\widehat{\mathsf{A}}(\dagger)(\langle u_0, \dots, u_n \rangle) = u_0$, and
 $\widehat{\mathsf{A}}(\triangleright)$ is the total function from U^+ to U^* such that for each
 $\langle u_0, \dots, u_n \rangle \in \mathsf{U}^+$,
 $\widehat{\mathsf{A}}(\triangleright)(\langle u_0, \dots, u_n \rangle) = \langle u_1, \dots, u_n \rangle$.

I call \hat{S} the expanded species assignment function, and \hat{A} the expanded attribute interpretation function. For all elements of U in its domain, \hat{S} equals the species assignment function, S. In addition, \hat{S} assigns the quasi-sort echain to the empty chain, and nechain to each nonempty chain of entities. \hat{A} equals the attribute interpretation function, A, for all attributes in its domain; however its domain also includes \dagger and \triangleright , and its range includes partial functions whose domains are chains. This makes it possible to interpret the quasi-attributes \dagger and \triangleright as symbols for navigating through sequences of entities.

The presence of chains in the formalism can also be felt in the definition of variable assignments. I will use variable assignments in a fairly standard fashion. However, since variables will be used in the arguments of relational expressions, and since arguments of relations can be entities or chains of entities, variable assignments must provide the possibility of assigning entities of the universe as well as sequences of entities of the universe to variables:

Definition 6 For each signature Σ , for each Σ interpretation $I = \langle U, S, A, R \rangle$,

Ass_I =
$$\overline{U}^{\mathcal{VAR}}$$
 is the set of variable assignments in I

I call each element of Ass_1 a variable assignment in I. Elements of Ass_1 are functions which assign entities or chains of entities of the universe to variables.

Up to this point the definitions closely followed the definitions of RSRL presented in [Richter, 2004a, chapter 3]. The deviation from earlier versions of RSRL begins with the definition of the syntax of Σ boxes in DEFINITION 7. The idea behind Σ boxes is that they mimic the attribute value matrices of the literature.

Suppose the signature Σ_1 from Figure 1.1. Based on this signature, we can write a simplified version of the HEAD FEATURE PRINCIPLE of [Pollard and Sag, 1994]. The consequent of this HEAD FEATURE PRINCIPLE is an example of a Σ_1 box.³ \square is a variable.

³When we spell out the entire principle, the complete consequent of the HEAD FEATURE PRINCIPLE will turn out to be a Σ_1 AVM formula, although the Σ_1 box discussed here will remain the crucial part of the formula.

1.1. DESCRIPTION

(3)
$$\begin{bmatrix} phrase \\ CAT \\ HEAD \\ HEAD \\ H_DTR \\ \begin{bmatrix} cat \\ head \end{bmatrix} \\ CAT \\ \begin{bmatrix} cat \\ HEAD \\ HEAD \end{bmatrix} \\ \end{bmatrix}$$

Boxes may comprise logical connectives such as disjunction or negation. The other most frequent standard logical connectives, conjunction, implication and bi-implication, are added to our definition of boxes for uniformity of the language, but are normally not used inside the syntactic layer of boxes in HPSG. As we will see, these connectives occur in the outer layer of syntax at the level of AVM formulae, which I will define when I have finished defining boxes. Here are some examples of negation and disjunction in Σ_1 boxes, as they might be observed in actual grammars:

(4) a.
$$\begin{bmatrix} sign \\ CAT \begin{bmatrix} cat \\ HEAD \neg [verb] \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} sign \\ CAT \begin{bmatrix} cat \\ HEAD \begin{bmatrix} [verb] \lor [noun] \end{bmatrix} \end{bmatrix}$$

 Σ boxes exist in two varieties, as *untagged* and as *tagged* Σ boxes. Tagged Σ boxes are matrices immediately preceded by a variable (or the colon), such as the Σ_1 box \square [*head*] which nestles in the *untagged* Σ_1 box in (3).

Definition 7 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, \mathbb{BOX}^{Σ} , \mathbb{UBOX}^{Σ} and \mathbb{TBOX}^{Σ} are the smallest sets such that

$$\mathsf{TBO} \mathbb{X}^{\Sigma} \cup \mathbb{U} \mathsf{BO} \mathbb{X}^{\Sigma} \subseteq \mathsf{BO} \mathbb{X}^{\Sigma},$$

for each $\sigma \in \widehat{\mathcal{G}}$, for each $\alpha_1 \in \widehat{\mathcal{A}}$, ..., for each $\alpha_n \in \widehat{\mathcal{A}}$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, ..., for each $\beta_n \in \mathbb{BOX}^{\Sigma}$,

$$\begin{bmatrix} \sigma \\ \alpha_1 & \beta_1 \\ \vdots \\ \alpha_n & \beta_n \end{bmatrix} \in \mathbb{UBO}\mathbb{X}^{\Sigma},$$

for each $\beta \in \mathbb{BOX}^{\Sigma}$, $\neg \beta \in \mathbb{UBOX}^{\Sigma}$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$, $[\beta_1 \land \beta_2] \in \mathbb{UBOX}^{\Sigma}$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$, $[\beta_1 \lor \beta_2] \in \mathbb{UBOX}^{\Sigma}$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$, $[\beta_1 \to \beta_2] \in \mathbb{UBOX}^{\Sigma}$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$, $[\beta_1 \leftrightarrow \beta_2] \in \mathbb{UBOX}^{\Sigma}$, for each $v \in \mathcal{VAR}^{:}$, for each $\beta \in \mathbb{UBOX}^{\Sigma}$, $v \beta \in \mathbb{TBOX}^{\Sigma}$.

I will call each element of \mathbb{TBOX}^{Σ} a tagged Σ box, each element of \mathbb{UBOX}^{Σ} an untagged Σ box, and each element of \mathbb{BOX}^{Σ} a Σ box. Notice that $\mathbb{TBOX}^{\Sigma} \cap \mathbb{UBOX}^{\Sigma} = \emptyset$ and $\mathbb{TBOX}^{\Sigma} \cup \mathbb{UBOX}^{\Sigma} = \emptyset$ and $\mathbb{TBOX}^{\Sigma} \cup \mathbb{UBOX}^{\Sigma} = \mathbb{BOX}^{\Sigma}$. Untagged Σ boxes of the form [σ] and tagged Σ boxes of the form $v[\sigma]$, where v is a variable, I will call atomic Σ matrices. A Σ matrix need not be atomic. There can be any finite number of symbols of the expanded attribute set in vertical order under the symbol, σ , of the expanded sort set, each of them followed by a Σ box. I will call a Σ box of that form a(n) (untagged) Σ matrix. Note again that the colon may syntactically appear in the position of variables.

If I were entirely strict I would have to distinguish the symbols for the logical connectives of the outer layer of syntax from the logical connectives inside Σ matrices. However, since the AVM formulae which we will use in actual grammars will always be of a syntactically unambiguous form, no confusion can arise in practice, and I will omit a syntactic distinction between the symbols for the logical connectives in the two layers of syntax.

The purpose of DEFINITION 8 is to provide the syntactic means to conjoin Σ matrices to complex expressions, and to add constructs missing from the matrices, *viz.* quantifiers and relational expressions, which are necessary for expressing most HPSG principles. Two principles are shown in (5). They are again based on the signature Σ_1 . (5a) is a complete HEAD FEATURE PRINCIPLE. (5b) shows a CONSTITUENT ORDER PRINCIPLE.

Obviously all logical connectives in (5) are outside of the Σ_1 matrices which occur therein. Let us call a Σ matrix which is not embedded in another Σ matrix maximal. Then we can say that the formulae in (5) contain four maximal Σ_1 matrices. Each one begins with a colon, whose syntactic function in this context it is to produce a matrix which is also an AVM formula and can be conjoined with other AVM formulae by logical connectives, or which can be combined with quantifiers. DEFINITION 8 allows us to do just that: **Definition 8** For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, \mathbb{AVM}^{Σ} is the smallest set such that

for each
$$\beta \in \mathbb{TBOX}^{\Sigma}$$
, $\beta \in \mathbb{AVM}^{\Sigma}$,
for each $\rho \in \mathcal{R}$, for each $v_1 \in \mathcal{VAR}$, ..., for each $v_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$
 $\rho(v_1, \ldots, v_{\mathcal{AR}(\rho)}) \in \mathbb{AVM}^{\Sigma}$,
for each $v_1 \in \mathcal{VAR}^{\varepsilon}$, for each $v_2 \in \mathcal{VAR}^{\varepsilon}$, $v_1 = v_2 \in \mathbb{AVM}^{\Sigma}$,
for each $v \in \mathcal{VAR}$, for each $\kappa \in \mathbb{AVM}^{\Sigma}$, $\exists v \kappa \in \mathbb{AVM}^{\Sigma}$,
for each $v \in \mathcal{VAR}$, for each $\kappa \in \mathbb{AVM}^{\Sigma}$, $\forall v \kappa \in \mathbb{AVM}^{\Sigma}$,
for each $\kappa \in \mathbb{AVM}^{\Sigma}$, $\neg \kappa \in \mathbb{AVM}^{\Sigma}$,
for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$, $(\kappa_1 \wedge \kappa_2) \in \mathbb{AVM}^{\Sigma}$,
for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$, $(\kappa_1 \to \kappa_2) \in \mathbb{AVM}^{\Sigma}$,
for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$, $(\kappa_1 \to \kappa_2) \in \mathbb{AVM}^{\Sigma}$,
for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$, $(\kappa_1 \to \kappa_2) \in \mathbb{AVM}^{\Sigma}$,
for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$, $(\kappa_1 \to \kappa_2) \in \mathbb{AVM}^{\Sigma}$,

I will call each element of \mathbb{AVM}^{Σ} a Σ AVM formula. Every tagged Σ box is also an atomic Σ AVM formula. I will call each Σ formula of the form $\rho(v_1, \ldots, v_{\mathcal{AR}(\rho)})$ a relational Σ AVM formula, and each equation between elements of $\mathcal{VAR}^{:}$ a Σ AVM equation. Both relational Σ AVM formulae and Σ AVM equations are atomic Σ AVM formulae. Complex Σ AVM formulae can be formed with quantification and the standard logical connectives in the usual way. When working with a fixed signature, I will refer to Σ boxes and Σ AVM formulae, respectively.

DEFINITION 8 completes the syntactic definitions which we need to write grammars. Later we will add a number of notational conventions which will introduce a number of very convenient simplifications to the syntax. Before we do that, however, we must give our formulae a semantics.

In order to assign meaning to Σ boxes and Σ AVM formulae, I need a few auxiliary concepts. The purpose of DEFINITIONS 9–11 is to provide the notion of the set of *components* of an entity u in a universe U, which is needed in order to pin down the particular kind of existential and universal quantification found in HPSG. As argued at length in [Richter et al., 1999, Richter, 2004a], quantification in HPSG is not quantification over the entire universe of objects as in first order logic. It is quantification over the components of entities in the universe. In the Σ_1 interpretation in (2a) above, the entities labeled *nelist*, *uther, cat, noun* and the two entities labeled *elist* are the (proper) components of the entity labeled *word*. The entity labeled *noun* and the lower one of the two entities labeled *elist* are the (proper) components of the entity labeled *cat*. When we have defined the set of components of an entity in a Σ interpretation, we only need one additional definition, a standard definition of what it means to change a variable assignment function minimally with respect to one variable in its domain, in order to say what the expressions of our syntax mean.

It turns out to be convenient to use the Σ terms of the original RSRL syntax and their meaning functions to define componenthood in Σ interpretations. Σ terms will also be a useful tool in defining the semantics of Σ matrices. Therefore we introduce Σ terms:

Definition 9 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, \mathcal{T}^{Σ} is the smallest set such that

 $: \in \mathcal{T}^{\Sigma},$ for each $v \in \mathcal{VAR}, v \in \mathcal{T}^{\Sigma}, and$ for each $\alpha \in \widehat{\mathcal{A}}$ and each $\tau \in \mathcal{T}^{\Sigma}, \tau \alpha \in \mathcal{T}^{\Sigma}.$

I will call each element of \mathcal{T}^{Σ} a Σ term. A Σ term consists of either the reserved symbol ':' or a variable followed by a (possibly empty) string of symbols of the expanded attribute set. Σ terms are interpreted as follows:

Definition 10 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, for each Σ interpretation $I = \langle U, S, A, R \rangle$, for each ass $\in Ass_I, T_I^{ass}$ is the total function from \mathcal{T}^{Σ} to the set of partial functions from U to U such that for each $u \in U$,

$$T_{\mathsf{I}}^{ass}(:)(u)$$
 is defined and $T_{\mathsf{I}}^{ass}(:)(u) = u$,
for each $v \in \mathcal{VAR}$, $T_{\mathsf{I}}^{ass}(v)(u)$ is defined and $T_{\mathsf{I}}^{ass}(v)(u) = ass(v)$,
for each $\tau \in \mathcal{T}^{\Sigma}$, for each $\alpha \in \widehat{\mathcal{A}}$,

$$\begin{split} T_{\mathsf{I}}^{ass}(\tau\alpha)(u) & is \ defined \\ iff \ T_{\mathsf{I}}^{ass}(\tau)(u) & is \ defined \ and \ \widehat{\mathsf{A}}(\alpha)(T_{\mathsf{I}}^{ass}(\tau)(u)) \ is \ defined, \ and \\ if \ T_{\mathsf{I}}^{ass}(\tau\alpha)(u) & is \ defined \\ then \ T_{\mathsf{I}}^{ass}(\tau\alpha)(u) &= \widehat{\mathsf{A}}(\alpha)(T_{\mathsf{I}}^{ass}(\tau)(u)). \end{split}$$

I will call T_1^{ass} the Σ term interpretation function with respect to I under a variable assignment in I, ass. Each Σ term is interpreted as a partial function from U to \overline{U} . The colon denotes the identity function in U. A variable, v, denotes a constant total function from U to \overline{U} , where each element of U is assigned the entity in U or the chain of entities in U that the variable assignment in I assigns to v. Finally, the meaning of complex Σ terms results from functional composition of the denotation of each of the symbols of the expanded attribute set succeeding the colon or variable in reverse order of their appearance in the Σ term, and the meaning of the colon or variable. Note that only Σ terms which begin with a variable which is followed by a (possibly empty) string of the reserved symbol \triangleright may map elements of U to chains of entities of U. All other terms denote functions which, if they are defined on any entity of the universe at all, may only map them to entities of U. Notice also that, in case τ begins with a variable, v, the value of $T_1^{ass}(\tau)(u)$ and the question of whether it is defined is independent of u, because it depends entirely on ass(v). I will say that a Σ term, τ , is *defined on an entity*, u, iff τ begins with the colon and $T_{\mathsf{I}}^{ass}(\tau)(u)$ is defined.

The Σ term interpretation functions T_1^{ass} will later be convenient for defining the meaning of Σ matrices, in particular for tagged Σ boxes. For pinning down the set of components of an entity u in an interpretation, Co_1^u , reference to Σ terms starting with the colon suffices:

Definition 11 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, for each Σ interpretation $I = \langle U, S, A, R \rangle$, and for each $u \in U$,

$$\mathsf{Co}_{\mathsf{I}}^{u} = \left\{ u' \in \mathsf{U} \middle| \begin{array}{l} \textit{for some ass} \in Ass_{\mathsf{I}}, \\ \textit{for some } \pi \in \mathcal{A}^{*}, \\ T_{\mathsf{I}}^{ass}(:\pi)(u) \textit{ is defined, and} \\ u' = T_{\mathsf{I}}^{ass}(:\pi)(u) \end{array} \right\}.$$

I will call $\mathsf{Co}^u_{\mathsf{I}}$ the set of components of u in I . Informally, DEFINITION 11 limits the set of components of an entity u in I to those entities in U which are accessible from u by Σ terms defined on u. The denotation of $:\pi$ is obviously independent of the variable assignment in I , since there is no variable in $:\pi$.

The last definition before we get to the meaning of Σ matrices is in fact just a notational convention for writing down variable assignments. It is formed after the well known notational conventions of any standard logic which comprises quantification:

Definition 12 For each signature Σ , for each Σ interpretation $I = \langle U, S, A, R \rangle$, for each ass $\in Ass_1$, for each $v \in \mathcal{VAR}$, for each $w \in \mathcal{VAR}$, for each $u \in \overline{U}$,

$$ass\frac{u}{v}(w) = \begin{cases} u & if \ v = w \\ ass(w) & otherwise. \end{cases}$$

For each variable assignment in I, ass, $ass \frac{u}{v}$ is just like the function ass, except that it assigns entity u to variable v.

Next I define the interpretation Ξ_{I}^{ass} for Σ boxes and Δ_{I}^{ass} for Σ AVM formulae. Following the syntactic structure of the logical languages I have to first define the meaning of Σ boxes, in order to be able to define the meaning of AVM formulae in which they may be contained.

What is interesting about the meaning of Σ boxes is that it is defined relative to Σ terms which lead to them. To see how this works, take the maximal Σ_1 matrix of the consequent of the HEAD FEATURE PRINCIPLE from Example (5a), repeated in (6a) for convenience:

(6) a. :
$$\begin{bmatrix} phrase \\ CAT \\ HEAD \\ HEAD \\ HDTR \\ \begin{bmatrix} cat \\ HEAD \\ CAT \\ CAT \\ HEAD \\ HEAD \\ HEAD \end{bmatrix} \end{bmatrix}$$

b.
$$\begin{bmatrix} sign \\ CAT \\ HEAD \end{bmatrix} \begin{bmatrix} cat \\ HEAD \end{bmatrix} \begin{bmatrix} cat \\ hEAD \end{bmatrix}$$
c.
$$\begin{bmatrix} cat \\ HEAD \end{bmatrix} \begin{bmatrix} head \end{bmatrix}$$

The Σ_1 boxes in (6b) and (6c) are both parts of the consequent of the HEAD FEATURE PRINCIPLE. In fact, the Σ_1 box in (6c) occurs twice in (6a). The meaning of the Σ_1 matrix in (6a) will be defined as being composed from the meanings of the Σ_1 matrices embedded in it, in combination with the meanings of the Σ_1 terms which lead to them. Thus the meaning of the Σ_1 matrix depicted in (6b), when interpreted as part of (6a), will be obtained by considering the term ': H_DTR' as well as the embedded matrix itself. The meaning of the matrix in (6c) will vary depending on which occurrence in (6a) we consider: The meaning of its first occurrence from the top will be obtained by considering the term ': CAT' plus the matrix; the meaning of its second occurrence will be obtained by interpreting the term ': H_DTR CAT' plus the matrix.

Logical connectives in boxes receive a classical interpretation, but always under the assumption that the term leading to the box is defined on the objects in the denotation of the expression.

1 Definition 13 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, for each Σ interpretation 2 $I = \langle U, S, A, R \rangle$, for each ass $\in Ass_{I}, \Xi_{I}^{ass}$ is the total function from the Cartesian product 3 of \mathcal{T}^{Σ} and \mathbb{BOX}^{Σ} to the power set of U such that

for each $\tau \in \mathcal{T}^{\Sigma}$, for each $\sigma \in \widehat{\mathcal{G}}$, for each $\alpha_1 \in \widehat{\mathcal{A}}$, ..., for each $\alpha_n \in \widehat{\mathcal{A}}$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, ..., for each $\beta_n \in \mathbb{BOX}^{\Sigma}$,

$$\Xi_{\mathsf{I}}^{ass}\left(\tau, \begin{bmatrix} \sigma \\ \alpha_{1} & \beta_{1} \\ \vdots \\ \alpha_{n} & \beta_{n} \end{bmatrix}\right) = \left\{ u \in \mathsf{U} \middle| \begin{array}{c} T_{\mathsf{I}}^{ass}(\tau)(u) \text{ is defined,} \\ \widehat{\mathsf{S}}(T_{\mathsf{I}}^{ass}(\tau)(u)) \stackrel{\frown}{\sqsubseteq} \sigma, \\ \text{for all } i \in \mathbb{N} \text{ such that } 1 \leq i \leq n, \\ T_{\mathsf{I}}^{ass}(\tau\alpha_{i})(u) \text{ is defined, and} \\ u \in \Xi_{\mathsf{I}}^{ass}(\tau\alpha_{i},\beta_{i}) \end{array} \right\}$$

for each $\tau \in \mathcal{T}^{\Sigma}$, for each $\beta \in \mathbb{BOX}^{\Sigma}$,

$$\Xi_{\mathsf{I}}^{ass}(\tau,\neg\beta) = \left\{ u \in \mathsf{U} \mid T_{\mathsf{I}}^{ass}(\tau)(u) \text{ is defined} \right\} \cap (\mathsf{U} \backslash \Xi_{\mathsf{I}}^{ass}(\tau,\beta)),$$

9 for each
$$\tau \in \mathcal{T}^{\Sigma}$$
, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$

$$\Xi_{\mathsf{I}}^{ass}(\tau, [\beta_1 \land \beta_2]) = \Xi_{\mathsf{I}}^{ass}(\tau, \beta_1) \cap \Xi_{\mathsf{I}}^{ass}(\tau, \beta_2),$$

for each
$$\tau \in \mathcal{T}^{\Sigma}$$
, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$,

$$\Xi_{\mathsf{I}}^{ass}(\tau, [\beta_1 \lor \beta_2]) = \Xi_{\mathsf{I}}^{ass}(\tau, \beta_1) \cup \Xi_{\mathsf{I}}^{ass}(\tau, \beta_2),$$

6

8

1.1. DESCRIPTION

for each
$$\tau \in \mathcal{T}^{\Sigma}$$
, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$,

$$\Xi_{\mathsf{I}}^{ass}(\tau, [\beta_1 \to \beta_2]) = (\mathsf{U} \setminus \Xi_{\mathsf{I}}^{ass}(\tau, \beta)) \cup \Xi_{\mathsf{I}}^{ass}(\tau, \beta_2),$$

15 for each $\tau \in \mathcal{T}^{\Sigma}$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$,

$$\Xi_{\mathsf{I}}^{ass}(\tau, [\beta_1 \leftrightarrow \beta_2]) = ((\mathsf{U} \setminus \Xi_{\mathsf{I}}^{ass}(\tau, \beta_1)) \cap (\mathsf{U} \setminus \Xi_{\mathsf{I}}^{ass}(\tau, \beta_2))) \cup (\Xi_{\mathsf{I}}^{ass}(\tau, \beta_1) \cap \Xi_{\mathsf{I}}^{ass}(\tau, \beta_2)),$$

and

for each $\tau \in \mathcal{T}^{\Sigma}$, for each $v \in \mathcal{VAR}^{:}$, for each $\beta \in \mathbb{UBOX}^{\Sigma}$,

$$\Xi_{\mathsf{I}}^{ass}(\tau, v \ \beta) = \left\{ u \in \mathsf{U} \ \left| \begin{matrix} T_{\mathsf{I}}^{ass}(\tau)(u) \ is \ defined, \\ T_{\mathsf{I}}^{ass}(\tau)(u) = ass(v), \ and \\ u \in \Xi_{\mathsf{I}}^{ass}(\tau, \beta) \end{matrix} \right\}.$$

I will call Ξ_{I}^{ass} the Σ box interpretation function with respect to I under a variable assignment in I , ass. The meaning of a Σ box is given by a function which maps a Σ term and the Σ box to a set of objects. The idea is that the term will be determined by how deeply the Σ box is nested in a maximal Σ matrix. We can think of the term as being the result of collecting all the attributes which must be traversed in wandering from the left bracket of the maximal Σ matrix to the Σ box in question, and concatenating them in the order in which they are encountered. Moreover, the term starts with the variable (or colon) which is used as the (obligatory) tag of the maximal Σ matrix.

 Ξ_1^{ass} is then defined in such a way that—if there is no negation involved, and τ starts with the colon—the Σ term must be defined on the objects denoted by the Σ box relative to the given term. Intervening negation symbols (or implication) may of course reverse the condition that a term leading to a nested Σ box must always be defined when determining the overall meaning of a maximal Σ matrix containing negation symbols. This can easily be seen by considering the effect of the set complement operation in line 8 of DEFINITION 13 upon the denotation of the nested Σ box β relative to the Σ term τ if β is a box which contains at least one attribute.

The final interesting case occurs in line 19, which concerns tagged Σ boxes. The denotation of a tagged Σ box relative to a Σ term τ is very similar to the corresponding untagged Σ box, except that here we see the additional condition that the object u' which the variable assignment in I assigns to the variable used as the tag must be the very same object in I which is also the result of applying the function named by τ to u. Loosely speaking, we may say that the tag must anchor the denotation of β exactly at the object u' pointed to by τ . If this construct occurs in the scope of negation, these conditions might of course change.

On the basis of the Σ box interpretation functions it is now straightforward to define the interpretation functions for Σ AVM formulae. For the interpretation of the atomic Σ AVM formulae which are also tagged Σ boxes, we can use the box interpretation functions by taking the tag of the box as the term argument and the remaining untagged box as the box argument of the interpretation function. Intuitively, the box will then be interpreted

19

14

16 17

18

relative to the object to which the tag points. For relational Σ AVM formulae, for Σ AVM equations and for quantificational expressions we can use the same definitions as for the description language of RSRL in [Richter, 2004a]. Note the restriction to components (and chains of components) of the denoted objects in the universe in the definition of quantification. This restriction reflects our earlier assumptions about the connectedness of the structures which our formal languages characterize. The logical constants are given a classical denotation using set complement, intersection and union:

Definition 14 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, for each Σ interpretation $I = \langle U, S, A, R \rangle$, for each ass $\in Ass_1$, Δ_I^{ass} is the total function from \mathbb{AVM}^{Σ} to the power set of U such that

for each $v \in \mathcal{VAR}^{:}$, for each $\beta \in \mathbb{UBOX}^{\Sigma}$,

$$\Delta^{ass}_{\rm I}(v \ \beta) = \Xi^{ass}_{\rm I}(v,\beta),$$

for each $\rho \in \mathcal{R}$, for each $v_1 \in \mathcal{VAR}, \ldots$, for each $v_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,

$$\Delta_{\mathsf{I}}^{ass}(\rho(v_1,\ldots,v_{\mathcal{AR}(\rho)})) = \Big\{ u \in \mathsf{U} \ \Big| \Big\langle ass(v_1),\ldots,ass(v_{\mathcal{AR}(\rho)}) \Big\rangle \in \mathsf{R}(\rho) \Big\},\$$

for each $v_1 \in \mathcal{VAR}^{:}$, for each $v_2 \in \mathcal{VAR}^{:}$,

$$\Delta_{\mathsf{I}}^{ass}(v_1 = v_2) = \left\{ u \in \mathsf{U} \mid T_{\mathsf{I}}^{ass}(v_1)(u) = T_{\mathsf{I}}^{ass}(v_2)(u) \right\}$$

for each $v \in \mathcal{VAR}$, for each $\kappa \in \mathbb{AVM}^{\Sigma}$,

$$\Delta_{\mathsf{I}}^{ass}(\exists v \; \kappa) = \left\{ u \in \mathsf{U} \middle| \begin{array}{l} \text{for some } u' \in \overline{\mathsf{Co}_{\mathsf{I}}^{u}}, \\ u \in \Delta_{\mathsf{I}}^{ass \frac{u'}{v}}(\kappa) \end{array} \right\}$$

for each $v \in \mathcal{VAR}$, for each $\kappa \in \mathbb{AVM}^{\Sigma}$,

$$\Delta_{\mathsf{I}}^{ass}(\forall v \; \kappa) = \left\{ u \in \mathsf{U} \middle| \begin{array}{l} for \; each \; u' \in \overline{\mathsf{Co}_{\mathsf{I}}^{u}}, \\ u \in \Delta_{\mathsf{I}}^{ass} \frac{u'}{v}(\kappa) \end{array} \right\},$$

for each $\kappa \in \mathbb{AVM}^{\Sigma}$,

$$\Delta_{\mathsf{I}}^{ass}(\neg\kappa) = \mathsf{U} \backslash \Delta_{\mathsf{I}}^{ass}(\kappa),$$

for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$,

$$\Delta_{\mathsf{I}}^{ass}((\kappa_1 \wedge \kappa_2)) = \Delta_{\mathsf{I}}^{ass}(\kappa_1) \cap \Delta_{\mathsf{I}}^{ass}(\kappa_2),$$

for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$,

$$\Delta_{\mathsf{I}}^{ass}((\kappa_1 \vee \kappa_2)) = \Delta_{\mathsf{I}}^{ass}(\kappa_1) \cup \Delta_{\mathsf{I}}^{ass}(\kappa_2),$$

for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$,

$$\Delta_{\mathsf{I}}^{ass}((\kappa_1 \to \kappa_2)) = \mathsf{U} \backslash \Delta_{\mathsf{I}}^{ass}(\kappa)) \cup \Delta_{\mathsf{I}}^{ass}(\kappa_2),$$

for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$, and

$$\Delta_{\mathsf{I}}^{ass}((\kappa_1 \leftrightarrow \kappa_2)) = ((\mathsf{U} \backslash \Delta_{\mathsf{I}}^{ass}(\kappa_1)) \cap (\mathsf{U} \backslash \Delta_{\mathsf{I}}^{ass}(\kappa_2))) \cup (\Delta_{\mathsf{I}}^{ass}(\kappa_1) \cap \Delta_{\mathsf{I}}^{ass}(\kappa_2)).$$

I will call Δ_{I}^{ass} the Σ AVM formula interpretation function with respect to I under a variable assignment in I, ass.

The meaning of *n*-ary relational Σ AVM formulae, $\rho(x_1, \ldots, x_n)$, depends on the sets of *n*-tuples of entities (and chains of entities) in the interpretation of the relation symbol ρ , $\mathsf{R}(\rho)$. Their meaning is simply a function of whether or not the given variable assignment in I is such that it assigns to the tuple of variables x_1 to x_n a tuple of entities (and chains of entities) which is in the interpretation of ρ . If the tuple of entities (and chains of entities) is in the interpretation of ρ , the relational expression denotes the entire universe of objects; otherwise it denotes the empty set. The denotation of relational Σ AVM formulae therefore depends entirely on the choice of the variable assignment in I, ass.

While this might seem strange at first, the reason for this definition becomes clear when we consider what we want to use our logical languages for. The description languages are supposed to give linguists the freedom to introduce any relations they might discover to be necessary for the description of human languages. Since we cannot anticipate the relations which linguists might want to use in the future, linguists have to be able to define the meaning of new relation symbols themselves within the formalism according to their needs. The resulting *relation principles* in their grammars will determine which entities must be in the denotation of the relation symbols in *models* of the grammars. These relation principles, just like all other grammatical principles, will be Σ AVM formulae *without free variables*. All variables in all principles will be bound by a quantifier. Under these circumstances the apparent arbitrariness of the meaning of relational Σ AVM formulae will vanish, since the relation principles; and the meaning of Σ AVM formulae without free variables will turn out to be independent of the choice of the variable assignment in l.

In order to prove this last claim, we first have to define an auxiliary function which, for each Σ AVM formula, gives us the sets of variables which occur free in it. We will need this function not only to prove our claim, but also to define the set of Σ AVM formulae without free variables. These we will need to define the notion of a grammar.

Definition 15 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$,

 $FV^{AVM}(:) = \emptyset,$

for each $v \in \mathcal{VAR}$, $FV^{AVM}(v) = \{v\}$,

for each $\tau \in \mathcal{T}^{\Sigma}$, for each $\alpha \in \widehat{\mathcal{A}}$, $FV^{AVM}(\tau \alpha) = FV^{AVM}(\tau)$,

for each $\sigma \in \widehat{\mathcal{G}}$, for each $\alpha_1 \in \widehat{\mathcal{A}}$, ..., for each $\alpha_n \in \widehat{\mathcal{A}}$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, ..., for each $\beta_n \in \mathbb{BOX}^{\Sigma}$,

$$FV^{AVM}\left(\left[\begin{array}{cc}\sigma\\\alpha_{1}&\beta_{1}\\\vdots\\\alpha_{n}&\beta_{n}\end{array}\right]\right) = \left\{v \in \mathcal{VAR} \mid \begin{array}{c} for \ some \ i \in \{1,\ldots,n\},\\v \in FV^{AVM}(\beta_{i})\end{array}\right\},$$

for each $\beta \in \mathbb{BOX}^{\Sigma}$, $FV^{AVM}(\neg \beta) = FV^{AVM}(\beta)$, for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$,

$$FV^{AVM}\left(\left[\beta_{1}\wedge\beta_{2}\right]\right)=FV^{AVM}\left(\beta_{1}\right)\cup FV^{AVM}\left(\beta_{2}\right),$$

for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$,

$$FV^{AVM}\left(\left[\beta_{1}\vee\beta_{2}\right]\right)=FV^{AVM}\left(\beta_{1}\right)\cup FV^{AVM}\left(\beta_{2}\right),$$

for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$,

$$FV^{AVM}\left(\left[\beta_{1} \rightarrow \beta_{2}\right]\right) = FV^{AVM}\left(\beta_{1}\right) \cup FV^{AVM}\left(\beta_{2}\right),$$

for each $\beta_1 \in \mathbb{BOX}^{\Sigma}$, for each $\beta_2 \in \mathbb{BOX}^{\Sigma}$,

$$FV^{AVM}\left(\left[\beta_1 \leftrightarrow \beta_2\right]\right) = FV^{AVM}\left(\beta_1\right) \cup FV^{AVM}\left(\beta_2\right),$$

for each $v \in \mathcal{VAR}^{:}$, for each $\beta \in \mathbb{UBOX}^{\Sigma}$,

$$FV^{AVM}\left(v\;\beta\right) = FV^{AVM}(v) \cup FV^{AVM}\left(\beta\right),$$

for each $\rho \in \mathcal{R}$, for each $v_1 \in \mathcal{VAR}, \ldots$, for each $v_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,

$$FV^{AVM}\left(\rho(v_1,\ldots,v_{\mathcal{AR}(\rho)})\right) = \{v_1,\ldots,v_{\mathcal{AR}(\rho)}\},\$$

for each $v_1 \in \mathcal{VAR}^:$, for each $v_2 \in \mathcal{VAR}^:$,

$$FV^{AVM}(v_1 = v_2) = FV^{AVM}(v_1) \cup FV^{AVM}(v_2),$$

for each $v \in \mathcal{VAR}$, for each $\kappa \in \mathbb{AVM}^{\Sigma}$,

$$FV^{AVM}(\exists v \ \kappa) = FV^{AVM}(\kappa) \setminus \{v\},\$$

for each $v \in \mathcal{VAR}$, for each $\kappa \in \mathbb{AVM}^{\Sigma}$,

$$FV^{AVM}(\forall v \ \kappa) = FV^{AVM}(\kappa) \setminus \{v\},\$$

for each $\kappa \in \mathbb{AVM}^{\Sigma}$, $FV^{AVM}(\neg \kappa) = FV^{AVM}(\kappa)$, for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$,

$$FV^{AVM}\left((\kappa_1 \wedge \kappa_2)\right) = FV^{AVM}(\kappa_1) \cup FV^{AVM}(\kappa_2)$$

for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$,

$$FV^{AVM}\left(\left(\kappa_1 \vee \kappa_2\right)\right) = FV^{AVM}(\kappa_1) \cup FV^{AVM}(\kappa_2),$$

for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$,

$$FV^{AVM}\left((\kappa_1 \to \kappa_2)\right) = FV^{AVM}(\kappa_1) \cup FV^{AVM}(\kappa_2),$$

for each $\kappa_1 \in \mathbb{AVM}^{\Sigma}$, for each $\kappa_2 \in \mathbb{AVM}^{\Sigma}$,

$$FV^{AVM}((\kappa_1 \leftrightarrow \kappa_2)) = FV^{AVM}(\kappa_1) \cup FV^{AVM}(\kappa_2).$$

For each signature Σ , FV^{AVM} is defined for Σ terms, Σ boxes and Σ AVM formulae and assigns a set of variables to each of them. I will say that the variables in the set assigned to an expression by FV^{AVM} occur free in that expression.

As we have already seen above, the meaning of arbitrary expressions may depend on the choice of the variable assignment in I under which an expression is evaluated. PROPOSI-TION 1 states that if two variable assignments in I agree on the values which they assign to the free variables in a Σ term, a Σ box or a Σ AVM formula, then the term, box or formula denote respectively the same partial function (in the case of terms) or set of objects (in the case of boxes and formulae) under the two variable assignments in I:

Proposition 1 For each signature Σ , for each Σ interpretation I, for each $ass_1 \in Ass_1$, for each $ass_2 \in Ass_1$,

for each $\tau \in \mathcal{T}^{\Sigma}$,

if for each
$$v \in FV^{AVM}(\tau)$$
, $ass_1(v) = ass_2(v)$
then $T_{\mathsf{I}}^{ass_1}(\tau) = T_{\mathsf{I}}^{ass_2}(\tau)$,

for each $\tau \in \mathcal{T}^{\Sigma}$, for each $\beta \in \mathbb{BOX}^{\Sigma}$,

if for each
$$v \in (FV^{AVM}(\tau) \cup FV^{AVM}(\beta))$$
, $ass_1(v) = ass_2(v)$
then $\Xi_{\mathsf{I}}^{ass_1}(\tau, \beta) = \Xi_{\mathsf{I}}^{ass_2}(\tau, \beta)$, and

for each $\kappa \in \mathbb{AVM}^{\Sigma}$,

if for each
$$v \in FV^{AVM}(\kappa)$$
, $ass_1(v) = ass_2(v)$
then $\Delta_{\mathsf{I}}^{ass_1}(\kappa) = \Delta_{\mathsf{I}}^{ass_2}(\kappa)$.

This result will become important for the meaning of grammatical principles. Grammatical principles are always Σ AVM formulae in which no variable occurs free. Since I will often need to refer to this set of formulae, I will introduce a distinguishing notation and terminology for it: Let \mathbb{AVM}_0^{Σ} be the set of Σ AVM formulae without free variables. I will call each element of \mathbb{AVM}_0^{Σ} a Σ AVM description. When working with a fixed signature, I will simply refer to them as AVM descriptions:

Definition 16 For each signature Σ ,

$$\mathbb{AVM}_0^{\Sigma} = \{ \kappa \in \mathbb{AVM}^{\Sigma} \mid FV^{AVM}(\kappa) = \emptyset \}.$$

As a corollary of PROPOSITION 1 we immediately get the following result:

Corollary 1 For each signature Σ , for each $\kappa \in \mathbb{AVM}_0^{\Sigma}$, for each Σ interpretation I, for each $ass_1 \in Ass_1$, for each $ass_2 \in Ass_1$,

$$\Delta_{\mathsf{I}}^{ass_1}(\kappa) = \Delta_{\mathsf{I}}^{ass_2}(\kappa).$$

According to COROLLARY 1, the meaning of Σ AVM descriptions is independent of the choice of the variable assignment in I. Thus in Σ AVM description interpretation functions with respect to I we can ignore particular variable assignments:

Definition 17 For each signature Σ , for each Σ interpretation $I = \langle U, S, A, R \rangle$, Δ_I is the total function from \mathbb{AVM}_0^{Σ} to the power set of U such that for each $\kappa \in \mathbb{AVM}_0^{\Sigma}$,

$$\Delta_{\mathsf{I}}(\kappa) = \left\{ u \in \mathsf{U} \middle| \begin{array}{l} \text{for each ass} \in Ass_{\mathsf{I}}, \\ u \in \Delta_{\mathsf{I}}^{ass}(\delta) \end{array} \right\}.$$

I will call Δ_{I} the Σ AVM description interpretation functions with respect to I. As one would expect, this is entirely parallel to the original denotation function for descriptions in the original syntax of RSRL.

Before we move on to the meaning of sets of Σ AVM descriptions (and thereby to the meaning of entire grammars), I will briefly discuss a few illuminating examples. For this purpose I will repeat the Σ_1 interpretation from (2b), and introduce five Σ_1 AVM descriptions. Henceforth I will call the interpretation depicted in (7a) I_1 , where $I_1 = \langle U_1, S_1, A_1, R_1 \rangle$. For simplicity I will sometimes call entities in U_1 entities in I_1 .





b. $: [cat] \lor : [elist]$

		word					
с.	:	PHON	$\begin{bmatrix} list \\ FIRST & [uther] \\ REST & [elist] \end{bmatrix}$				
		CAT	$\begin{bmatrix} cat \\ HEAD & [noun] \\ SUBCAT & [elist] \end{bmatrix}$				
d.	-	: [verb]]				
e.	$\exists 1 \ 1 \ [elist]$						
f.	\forall	1 1 el	list				

According to the Σ_1 AVM formula interpretation function with respect to I_1 which follows from DEFINITION 14, the description (7b) denotes the subset of those entities in U_1 which are in the denotation of : [*cat*] or in the denotation of : [*elist*]. Following this back to the Σ_1 box interpretation function with respect to I_1 as provided by DEFINITION 13, these are those elements of U_1 which are either of species *cat* or of species *elist* (line 6 of DEFINITION 13, with $\tau = :$). This is, however, the set of entities labeled by the integers 3, 4, 6 and 8 in (7a). This is of course what we would intuitively expect under a natural reading of : $[cat] \lor : [elist]$. It denotes the set of entities of sort *cat* or of sort *elist* in the given interpretation.

(7c) is a much more complex Σ_1 box in the sense that here one maximal matrix contains six non-maximal matrices. In order to determine the denotation of (7c) in I_1 , line 6 of DEFINITION 13 is again crucial, this time including the recursivity of the Σ_1 box interpretation function for nested boxes. According to this definition, (7c) denotes the subset of all entities in U_1 such that they are of sort *word*, the terms :PHON, :PHON FIRST, :PHON REST, :CAT, :CAT HEAD, :CAT SUBCAT, are defined on them, and their values on each entity in question are labeled respectively by subsorts of *list*, *uther*, *elist*, *cat*, *noun*, and *elist*. The only entity in I_1 which fulfills all of these requirements is the *word* entity labeled 0.

(7d) is a very simple description. According to DEFINITION 14 and DEFINITION 13 it denotes the set complement of the set of those entities in I_1 which are in the denotation of : [*verb*]; or, in other words, the set of all entities in I_1 which are not of species *verb*. (7d) thus denotes the set of all entities in I_1 , with the exception of the one with label 9.

(7e) is an example of existential quantification. According to the clause concerning existential quantification in DEFINITION 14, $\exists \exists \exists \exists [elist]$ denotes the set of entities, u, in U_1 such that there is a component u' of u which is in the denotation of $\exists [elist]$, with ass assigning u' to $\exists denotes de$

(7f) illustrates the meaning of universal quantification in RSRL. In contrast to existential quantification, the universally quantified expression denotes the set of all entities in U_1 such that *all* of their components are of sort *elist*. This yields the set of entities labeled 3 and 6.

We can now use the Σ AVM description interpretation functions with respect to I to determine the meaning of sets of Σ AVM descriptions, which we will need to determine the meaning of grammars with sets of grammatical principles. Collections of grammatical principles will of course be formulated as sets of AVM descriptions, just like in symbolizations of the grammar of English in [Pollard and Sag, 1994]. I will call sets of Σ AVM descriptions a Σ *theory*. The idea is that Σ theories denote those objects in Σ interpretations which are described by every AVM description in the theory:

Definition 18 For each signature Σ , for each Σ interpretation $I = \langle U, S, A, R \rangle$, Θ_{I}^{AVM} is the total function from the power set of $\mathbb{AVM}_{0}^{\Sigma}$ to the power set of U such that for each $\theta \subseteq \mathbb{AVM}_{0}^{\Sigma}$,

$$\Theta_{\mathsf{I}}^{AVM}(\theta) = \left\{ u \in \mathsf{U} \middle| \begin{array}{c} \text{for each } \kappa \in \theta, \\ u \in \Delta_{\mathsf{I}}(\kappa) \end{array} \right\}.$$

I will call $\Theta_{\mathsf{I}}^{AVM}$ the Σ theory denotation function with respect to I . Before we take a closer look at this, we will define what a grammar is:

Definition 19 Γ is a grammar iff

$$\begin{split} \Gamma \ is \ a \ pair \ \langle \Sigma, \theta \rangle, \\ \Sigma \ is \ a \ signature, \ and \\ \theta \subseteq \mathbb{AVM}_0^{\Sigma}. \end{split}$$

A grammar is simply a pair consisting of a signature Σ and a Σ theory.

We are now in a position to write a small grammar based on our small signature Σ_1 from Figure 1.1 on page 21. The principles of this grammar are modeled according to the principles of actual HPSG grammars, but everything is kept extremely simple. The theory of our small grammar, $\langle \Sigma_1, \theta_1 \rangle$, contains only five Σ_1 AVM descriptions, the principles of the grammar. They are enumerated in (8). Each principle is assigned a name reminiscent of corresponding principles in more comprehensive grammars. Each principle is preceded by its formulation in natural language.

(8) a. WORD PRINCIPLE:

There are two words in the grammar: The noun *Uther* and the intransitive verb *walks*, which takes a noun as its argument.

:	$: [word] \rightarrow$									
1	/				word					1)
						list	7			
		word	-]	PHON	FIRST w	alks]			
			list			REST [el	list]			
		PHON	FIRST [uther]			cat	-		-	1
	:		REST [elist]	\vee :		HEAD	[verb]			
			cat				[list]		1	
		CAT	HEAD [noun]		CAT			cat	7	
			SUBCAT [elist]			SUBCAT	FIRST	HEAD	[noun]	
		_		-				SUBCAT	[elist]	
						L	REST	$\left[elist\right]$]] /

b. ID PRINCIPLE:

In each phrase the single element on the SUBCAT list of the head daughter is identical with the CAT value of the non-head daughter, and the entire phrase has an empty SUBCAT list.⁴

 $^{^4{\}rm The}$ consequent of this ID PRINCIPLE is very similar to a HEAD SUBJECT SCHEMA, with a SUBCATEGORIZATION PRINCIPLE built in.

$$: [phrase] \rightarrow \exists \mathbf{I} : \begin{bmatrix} phrase \\ CAT & \begin{bmatrix} cat \\ SUBCAT & [elist] \end{bmatrix} \\ H-DTR & \begin{bmatrix} sign \\ CAT & \begin{bmatrix} cat \\ SUBCAT & \begin{bmatrix} list \\ FIRST & \mathbf{I} & [cat] \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ NH-DTR & \begin{bmatrix} sign \\ CAT & \mathbf{I} & [cat] \end{bmatrix} \end{bmatrix}$$

c. Head Feature Principle:

In each phrase, the HEAD values of the phrase and of the head daughter are identical.

$$: [phrase] \rightarrow \exists \mathbf{1} : \begin{bmatrix} phrase \\ CAT \\ HEAD \end{bmatrix} \begin{bmatrix} cat \\ HEAD \end{bmatrix} \begin{bmatrix} head \end{bmatrix} \\ H_DTR \begin{bmatrix} sign \\ CAT \\ HEAD \end{bmatrix} \begin{bmatrix} cat \\ HEAD \end{bmatrix} \end{bmatrix}$$

d. Constituent Order Principle:

In each phrase, the phonology of the mother is the concatenation of the phonologies of the non-head daughter and the head daughter, in this order.

$$: [phrase] \rightarrow \exists 1 \exists 2 \exists 3 \qquad \left(\begin{array}{c} \left[\begin{array}{c} phrase \\ PHON & \exists [list] \\ H_DTR & \left[\begin{array}{c} sign \\ PHON & 2 & [list] \\ \end{array} \right] \\ NH_DTR & \left[\begin{array}{c} sign \\ PHON & 1 & [list] \\ \end{array} \right] \right) \land append(1, 2, 3) \\ \end{array} \right)$$

e. Append Principle:

An entity, u, in an interpretation, I, is described by the APPEND PRINCIPLE if and only if for each triple, $\underline{\square}$, $\underline{\square}$, $\underline{\square}$, $\underline{\square}$, of components of u, one of the following two conditions holds: (1) $\underline{\square}$ is of sort *elist*, $\underline{\square}$ is of sort *list* and $\underline{\square}$ is identical with $\underline{\square}$, or (2) the first element on list $\underline{\square}$ is identical with the first element on list $\underline{\square}$, and the rest of list $\underline{\square}$, the list $\underline{\square}$ and the rest of list $\underline{\square}$ are also in the **append** relation of I.

1.1. DESCRIPTION

$$\left(\begin{array}{c} \texttt{append}(\underline{1},\underline{2},\underline{3}) \leftrightarrow \left(\begin{array}{c} (\underline{1}[\textit{elist}] \land \underline{2}[\textit{list}] \land \underline{2} = \underline{3}) \lor \\ \exists \underline{4} \exists \underline{5} \exists \underline{6} \left(\begin{array}{c} \underline{1}[\textit{elist}] \land \underline{2} = \underline{3} \right) \lor \\ \exists \underline{4} \exists \underline{5} \exists \underline{6} \left(\begin{array}{c} \underline{1}[\textit{list}] & \underline{1} \\ \texttt{FIRST} & \underline{4}[\textit{top}] \\ \texttt{REST} & \underline{5}[\textit{list}] \end{array} \right) \land \underline{3} \begin{bmatrix} \textit{list} \\ \texttt{FIRST} & \underline{4}[\textit{top}] \\ \texttt{REST} & \underline{6}[\textit{list}] \end{bmatrix} \\ \land \texttt{append}(\underline{5},\underline{2},\underline{6}) \end{array} \right) \right) \right)$$

When we check the denotation of the set of descriptions consisting of the the principles (8a)-(8d) of the grammar $\langle \Sigma_1, \theta_1 \rangle$ in the interpretation I_1 , we find that it denotes the entire universe, U_1 . As far as the principles (8b)-(8d) are concerned, this is entirely trivial: since there is no entity labeled *phrase* in I_1 , the negation of the antecedents (which is : [*phrase*] in all three principles) denotes the entire universe of entities. Only the WORD PRINCIPLE is slightly more interesting: since there is an entity labeled *word* in I_1 , this entity (0) must also be in the denotation of one of the disjuncts in the consequent of the WORD PRINCIPLE. But, as we have already seen in example (7c), this is the case for the first disjunct, which is identical to (7c). Therefore the WORD PRINCIPLE also denotes the entire universe of entities in I_1 .

The situation becomes more complicated when we also consider the APPEND PRINCI-PLE, (8e). For the APPEND PRINCIPLE to be true of an entity u, all triples of components of u which are in the denotation of the variables \square , \square and \square in a disjunct of the formula to the right of the bi-implication symbol must, in the case that the formula is true of an entity, be in the denotation of **append**. Conversely, for the APPEND PRINCIPLE to be true of an entity u, all triples of components of u which are in the denotation of **append** must be in the denotation of the variables \square , \square and \square in a disjunct of the formula to the right of the bi-implication.

In the given interpretation, I_1 , only the triple $\langle 8, 8, 8 \rangle$ is in append.⁵ Evidently this means that all u in I_1 which have entity 8 as a component are not in the denotation of the APPEND PRINCIPLE, since entity 8, being of species *cat*, cannot be in the denotation of a single matrix tagged with \Box , \Box and \exists in the conjuncts (of the two disjuncts) on the right hand side of the bi-implication. This means that entity 8 is not in the denotation of the APPEND PRINCIPLE, because entity 8, as any other entity in a universe, is a component of itself.

Inspecting the formula ϕ to the right of the bi-implication symbol reveals, furthermore, that among other triples, triples with the same *elist* entity u' must be in **append** for any entity u with component u', in order to possibly be in the denotation of the APPEND PRINCIPLE. This is so because the first disjunct of ϕ and the bi-implication ensure the following: if \Box , \Box and \exists refer to the same *elist* component u' of an entity, then $\langle u', u', u' \rangle$ must be in $\mathsf{R}(\texttt{append})$. But this requirement excludes 0, 1, 3, 4, 6 and also 8 from being in the denotation of the APPEND PRINCIPLE.

What about the remaining entities, 7, 9, 5, and 2? Since none of them are labeled by a maximal specific subsort of *list*, we can show that none of them can be in the denotation of

 $^{^{5}}$ To be more precise, only the triple of the entity labeled 8 is in append.

a matrix tagged $\underline{1}$, $\underline{2}$ or $\underline{3}$ in a conjunct to the right of the bi-implication symbol. At the same time, none of them occur in a triple in **append** in I_1 . As we have already observed, $\langle 8, 8, 8 \rangle$ is the only triple in **append**. The set of entities 2, 5, 7, and 9 is thus the set of entities in the denotation of the APPEND PRINCIPLE.

From the argument above we conclude that our grammar $\langle \Sigma_1, \theta_1 \rangle$ denotes the set of entities 2, 5, 7 and 9 in U₁ under the theory denotation function with respect to I₁.

We do not only want to know which entities in an arbitrary interpretation are in the denotation of our grammars. Clearly, most interpretations of signatures will be uninteresting, and determining which entities in them are in the denotation of a given grammar is not the ultimate purpose of writing grammars. With grammars we want to characterize certain interpretations as well-formed with respect to our theory of language: these interpretations should behave as our theory predicts, and (some of) them are arguably the ones which we are interested in. A minimal requirement pointing in this direction is that a relevant interpretation be a *model* of a grammar:

Definition 20 For each grammar $\Gamma = \langle \Sigma, \theta \rangle$, for each Σ interpretation $I = \langle U, S, A, R \rangle$,

I is a Γ model iff $\Theta_{I}^{AVM}(\theta) = U$.

In $\langle \Sigma, \theta \rangle$ models $\langle U, S, A, R \rangle$, each entity in the universe of entities, U, is in the denotation of each principle of grammar in θ . In light of the idea that linguistic structures are connected collections of entities under a topmost entity in interpretations, the notion of a Γ model expresses a requirement to the effect that each entity in each structure must obey all principles of grammar. The requirement that an interesting interpretation be a model of a grammar is a minimal requirement necessary for characterizing those interpretations which linguists intend to describe with their grammars. For example, relation principles such as the APPEND PRINCIPLE will be obeyed by models of the grammar $\langle \Sigma_1, \theta_1 \rangle$, thus giving the relation symbol append the meaning which it should have and which is necessary in order to give the relational formula in the CONSTITUENT ORDER PRINCIPLE the intended restrictive force. As soon as append no longer has a meaning consistent with the APPEND PRINCIPLE in an interpretation, i.e. as soon as it is no longer a model of the APPEND PRINCIPLE, even the fact that the interpretation is a model of the CONSTITUENT ORDER PRINCIPLE no longer guarantees that the phonology of a phrase is a concatenation of the phonology of its head daughter and its non-head daughter, because append might have an unexpected denotation.

In the next section we will investigate further conditions on models in order to narrow down the class of structures which are good candidates for the meaning of grammars. In this discussion we will also introduce assumptions about the structure of language which are typically embodied in HPSG grammars, and we will consider the intended empirical relevance of models of grammars. This will be of immediate consequence for the purposes and status which we can ascribe to the semantic structures in the denotation of RSRL grammars.

Before we turn to a first examination of the meaning of grammars, we will discuss a final example of a relation principle. Equipped with the notion of models of grammars, we are now in a position to see that relation principles which fix the meaning of relation symbols do indeed lead to intuitively correct denotations of relation symbols in models of grammars. A first example of this kind of relation principle was the APPEND PRINCIPLE in example (8e), although we did not consider its denotation in models of $\langle \Sigma_1, \theta_1 \rangle$.

The MEMBER PRINCIPLE in (9) is significantly simpler than the previous relation principle, the APPEND PRINCIPLE of (8e). With the MEMBER PRINCIPLE we mean to express the relation of list membership. Intuitively, an entity is a member of a list entity just in case it occurs somewhere on the list under the list entity in question; and every element on a list under a list entity must be in the membership relation to every relevant list entity. Of course, the property of list membership is always formulated with respect to the encoding of lists through certain configurations of entities in the universe of interpretations. Under our signature Σ_1 , the property of occurring on a list must be expressed using the sort symbols *list, nelist* and *elist* and the attributes FIRST and REST, which provide access to the first element on a list and the tail of that list.⁶

The MEMBER PRINCIPLE fulfills all of the conditions we have just discussed:

(9) Member Principle:

$$\forall \exists \forall \exists \left(\operatorname{member}(\underline{1},\underline{2}) \leftrightarrow \left(\begin{array}{c} 2 \begin{bmatrix} list \\ FIRST \ \underline{1} \ [top] \end{bmatrix} \lor \\ \exists \exists \left(2 \begin{bmatrix} list \\ REST \ \underline{3} \ [top] \end{bmatrix} \land \operatorname{member}(\underline{1},\underline{3}) \right) \end{array} \right) \right)$$

Now assume that our second grammar, $\langle \Sigma_1, \theta_2 \rangle$, is just like our first grammar, except that we add the MEMBER PRINCIPLE to θ_1 in order to obtain θ_2 .

The Σ_1 interpretation $I_2 = \langle U_2, S_2, A_2, R_2 \rangle$ in (10) is a $\langle \Sigma_1, \theta_2 \rangle$ model.

⁶In Section 3.1 we will have the opportunity to briefly investigate the abstract properties of parts of signatures which encode lists in RSRL grammars.

(10) A picture of interpretation I_2 :



$$\begin{split} \mathtt{member} &= \{ \langle 6, 2 \rangle, \langle 7, 2 \rangle, \langle 7, 3 \rangle, \langle 8, 2 \rangle, \langle 8, 3 \rangle, \langle 8, 4 \rangle \} \\ \mathtt{append} &= \begin{cases} \langle 2, 5, 2 \rangle, \langle 3, 5, 3 \rangle, \langle 4, 5, 4 \rangle, \langle 5, 2, 2 \rangle, \langle 5, 3, 3 \rangle, \langle 5, 4, 4 \rangle, \langle 5, 5, 5 \rangle, \\ \langle 5, 11, 11 \rangle, \langle 5, 12, 12 \rangle, \langle 11, 2, 2 \rangle, \langle 11, 3, 3 \rangle, \langle 11, 4, 4 \rangle, \langle 11, 5, 5 \rangle, \\ \langle 11, 11, 11 \rangle, \langle 11, 12, 12 \rangle, \langle 12, 2, 2 \rangle, \langle 12, 3, 3 \rangle, \langle 12, 4, 4 \rangle, \langle 12, 5, 5 \rangle, \\ \langle 12, 11, 11 \rangle, \langle 12, 12, 12 \rangle \end{cases}$$

The interpretation I_2 is a model of the MEMBER PRINCIPLE because each element on the lists, represented as the FIRST values of *nelist* entities, is in the **member** relation with all *nelist* entities preceding the *nelist* entity whose FIRST value it is; and each element is in the **member** relation with the *nelist* entity whose FIRST value it is. With the characterization that '*nelist* entities precede an *nelist* entity,' *u*, I designate the *nelist* entities from which *u* can be reached following a sequence of REST arcs. For example, the *nelist* entity 2 precedes the *nelist* entities 3 and 4 in I_2 .

The reader may also confirm that I_2 is a model of θ_1 , because all entities in U_2 are in the denotation of each principle of θ_1 . This is particularly easy to see for the principles

(8a)–(8d), since all entities are in the denotation of the negation of the antecedents of these principles. There are simply no words or phrases in this model of the grammar. It is much more difficult to verify for the APPEND PRINCIPLE. The **append** relation which is a model of the APPEND PRINCIPLE is only given here to illustrate a very common although non-trivial relation principle. At the same time it may remind the reader of the real complexity of apparently innocuous HPSG principles.

1.2 Meaning

Models of RSRL grammars consist of configurations of entities which, if the symbolization of the grammar does not contain mistakes, agree with the generalizations of the grammar writer about the shape of expressions in the natural language: every entity in a model is such that all linguistic generalizations, expressed as a set of principles of grammar, are true of it. This is the reason why models immediately seem to be good candidates for structures in which the predictions of a grammar can be observed.

However, it is easy to see that models are not uniquely determined by grammars. Any realistic grammar of a natural language has a huge number of, or infinitely many, models; and each pair of models usually comprises non-isomorphic configurations of entities. It is very easy to see that these models differ with respect to their configurations of entities in ways which are crucial for the linguist. One model of a correct grammar of English may contain a configuration for the sentence *Mary is sleeping*, whereas another may not. As long as we do not introduce additional assumptions about relevant models, a model may contain one or more sentences, words, or well-formed component configurations thereof. To take our grammar $\langle \Sigma_1, \theta_2 \rangle$ from the previous section as an example, we may note that the $\langle \Sigma_1, \theta_2 \rangle$ model I₂ presented in (10) does not even contain the single sentence *Uther walks*, which is licensed by this grammar; neither does I₂ contain the two words *Uther* or *walks*, which are described in the lexicon of $\langle \Sigma_1, \theta_2 \rangle$. It is a model containing no sign at all.

An arbitrarily chosen single model obviously does not necessarily reflect the entirety of the predictions of a grammar. The meaning of a grammar cannot thus be given in terms of an arbitrary model. In addition, there are empirically significant questions about a grammar which cannot be answered by choosing just any model. For example, a very natural question for a linguist to ask is the question whether a grammar *overlicenses* or *underlicenses* a given natural language. Following [Richter, 2004a, p. 90f.], these two terms are formed in analogy with the two terms *overgeneration* and *undergeneration*, which are known from traditional generative grammar. Given our notion of a model of a grammar, we can explain their analogues 'overlicensing' and 'underlicensing' as follows: A grammar *overlicenses* just in case there are models which contain configurations of entities (taken to be utterances) which should not be in the language, because they are ungrammatical. It underlicenses just in case the grammar does not license configurations of entities (again taken to be utterances) which are deemed to be in the language, since they are judged to be grammatical expressions of the language by every native speaker of the language.

In the explanation of overlicensing and underlicensing, we quantified over the collec-

tions of all models of grammars. The question arises whether it is possible to pin down either a single model or an appropriate class of models which could be used instead. These 'intended' models should have the property of containing everything we know about the configurations of a language relative to a given grammar. Such models would be interesting because they would reliably reveal more properties of a grammar than the arbitrary models we considered before. If we can characterize models which embody the entirety of the predictions of a grammar in general terms, we gain access to models whose investigation could lead to the discovery of interesting facts about the structure of natural languages, especially by a comparative study of models of different languages. A precise characterization of the 'intended' models of a grammar might also be a good starting point for stating a meaningful and strong necessary condition for what it means for a grammar to be true of a natural language. The formulation of such a condition can be achieved by clarifying the relationship between the intended models and the natural languages which linguists want to describe with their grammars.

The simplest way of obtaining a model comprising all the predictions of a grammar seems to be to take a model comprising all possible models of a grammar, because it should contain everything which a given grammar can possibly describe. However, there are immediate unfortunate consequences which make this move undesirable. On the technical side, the described model would no longer be a set but a proper class. On the conceptual side, the model would indeed contain every entity for which the grammar is true according to its configurational properties. This will include all kinds of non-linguistic entities, including—to give a concrete example—models built from mathematical entities.⁷ However, these non-linguistic entities cannot be excluded from models through definitions, because there is no well-defined property of being linguistic. The large model of all possible models would thus be a bad candidate for a theory of grammatical truth in linguistics.

Paul King was the first to note and discuss the problem of how to capture the empirically important notion of truth of a grammar with respect to a natural language in the context of a mathematically rigorous HPSG formalism.⁸ Using a characterization which slightly differed from the one I gave, King already pointed out the unwanted consequences of assuming the comprehensive maximal model I sketched above. He studied a sequence of apparently plausible interim hypotheses which formulated different necessary conditions for what it means for a grammar to be true of a language, and thus revealed the shortcomings of various seemingly reasonable options for formulating this kind of condition. As a result of his careful study, [King, 1999, p. 343] proposed the following condition of grammatical truth, formulated in three interdependent sub-conditions:

A grammar $\langle \Sigma, \theta \rangle$ is true of a natural language only if

⁷Mathematical models of (R)SRL grammars are known from the investigation of model-theoretic properties of grammars in the literature. They are typically built as canonical models which have a particularly transparent structure. The existing proposals are summarized and compared in [Richter, 2004a].

⁸In [King, 1999]. King's discussion is technically precise in terms of the SRL (Speciate Re-entrant Logic) formalism. All relevant properties of SRL carry over to RSRL. Moreover, each SRL grammar can be expressed as an RSRL grammar. See [Richter, 2004a] for discussion.

- 1. the natural language can be construed as an interpretation, I, of the signature Σ , where I is a system of possible linguistic tokens,
- 2. each description in θ is true of each entity in the natural language, i.e., the natural language, I, is a $\langle \Sigma, \theta \rangle$ model, and
- 3. some description in θ is false for a component of each entity, u', in any Σ interpretation, I', for which no entity u in the natural language, I, has isomorphically configured components.

To a large extent Conditions 1 and 2 only express assumptions which are inherent already in choosing RSRL as the formalism for formulating generalizations about languages. The only statement in these two conditions which goes beyond that is King's assumption that the interpretations of interest are systems of *possible tokens*. This stands in sharp contrast to the more widespread idea that models of linguistic grammars should be understood as systems of *types*. It was this aspect of King's theory which led to objections to King's theory of linguistic truth. For the moment I will not take a position on the difficult issue of the type-token debate, and I will postpone comments on the choice between types and tokens as adequate objects of the study of language until we have investigated a grammar and its models more closely. I will come back to this topic in Section 2.4.

Condition 3 finally goes well beyond what is already implicit in the structure of a formalism employing descriptions and their denotation. King's third condition introduces the crucial aspect of comprehensiveness to the requirements. It makes the complete collection of all possible models of a grammar relevant for the formation of the intended models. The essential idea is that the intended models collect images of the configurations in all possible models. Its objective is to make the intended models big enough so that they contain all possible shapes of configurations of entities which a grammar licenses, without necessarily containing all entities in the world which are configured in the right way. This is crucial in order to avoid the presence of non-linguistic entities in the interpretation. At the same time the formulation of the condition is liberal enough to obtain structures which can be understood as a system of possible tokens. This is achieved by not determining the possible number of isomorphic configurations in the intended models. King's formulation of the condition also avoids the unfortunate consequence of turning models which are big enough to capture all predictions of the grammar into a proper class.

How the third condition works can best be seen by considering a slight reformulation of it. It can be re-stated equivalently as saying that whenever there is an entity u together with its configuration of components in a model of the grammar, then the intended model contains at least one isomorphic copy of u and its configuration of components. Restricting the requirement of completeness to requiring an undetermined number of isomorphic copies of each possible configuration (instead of requiring the presence of each and every appropriately shaped configuration) is King's way of avoiding non-linguistic entities in the intended models of grammars and of maintaining the set-size of models.

The class of models of a grammar whose shape complies with King's conditions is called the class of *exhaustive models* of the grammar. One straightforward possibility to define the exhaustive models of a grammar is shown in DEFINITION 21:

Definition 21 For each grammar $\Gamma = \langle \Sigma, \theta \rangle$, for each Σ interpretation I,

I is an exhaustive Γ model iff

I is a Γ model, and for each $\theta' \subseteq \mathbb{AVM}_0^{\Sigma}$, for each Σ interpretation I', if I' is a Γ model and $\Theta_{\mathsf{L}}^{AVM}(\theta') \neq \emptyset$ then $\Theta_{\mathsf{L}}^{AVM}(\theta') \neq \emptyset$.

An exhaustive Γ model, I, is a Γ model such that any theory which has a nonempty denotation in any other Γ model also has a nonempty denotation in I. This means that an exhaustive Γ model contains indiscernible counterparts to all configurations under an entity that occur in some model of Γ . We can say that a grammar is true of a natural language if the natural language is an exhaustive model of the grammar.

Instead of defining exhaustive models on the basis of the denotation of theories in models, it is possible to give an alternative algebraic definition in terms of congruent configurations of entities in different models. I have already used this algebraic characterization in the explanation of the intuitions behind exhaustive models above, because it is easier to picture than the definition based on descriptive indiscernibility. The algebraic definition might even be closer to the intuitions behind King's third criterion, because it appeals more directly to the idea of having all possible configurations of entities in exhaustive models. However, since the two characterizations are logically equivalent, since the algebraic definition needs a few additional preliminary steps, and since technical details are of no interest to our present discussion I will omit a definition of the algebraic characterization of exhaustive models.⁹

It can be shown that each grammar which has a nonempty model also has a nonempty exhaustive model. And, just as we observed for models, exhaustive models are not uniquely determined either. Most prominently, nothing in the definition of the exhaustive models of a grammar fixes the number of isomorphic copies of the same configuration of entities in the model. For each possible configuration there might be arbitrarily many isomorphic copies. On the other hand, additional restrictions on exhaustive models can in principle make each configuration structurally unique by excluding the presence of isomorphic copies.¹⁰ For this reason endorsing exhaustive models as the classes of intended models does not entail an irrevocable decision in favor of token models over type models. This basic openness of the concept of exhaustive models allows us to leave the controversial philosophical issue of token models as opposed to type models undecided for the moment until we have gathered more knowledge about the contents of exhaustive models of actual HPSG grammars.

⁹Readers interested in this may want to consult [Richter, 2004a, Section 2.2.2.2]. King's conditions are discussed in more detail in [Richter, 2004a, pp. 99–102]. Pollard's objections to possible tokens as the relevant entities subject to grammatical description are reported in [Richter, 2004a, p. 119f.]).

¹⁰Alternatively, if the reader is troubled by the prospect of having possible tokens instead of types in exhaustive models, he or she may simply think of each collection of isomorphic connected configurations of entities in an exhaustive model as an equivalence class which represents the desired type.

All the assumptions about the nature of human languages which come with exhaustive models are made explicit in King's condition of grammatical truth. Since exhaustive models are intrinsically non-committal—granting us the freedom to introduce slight changes through additional restrictions if necessary, and *pace* King's clear stance on the type-token issue—and since they do not introduce any algebraic or ontological restrictions on interpretations which might obscure facts about the denotation of grammars, I find them ideal for investigating HPSG grammars and their meaning.

In Chapter 2 I will investigate the question of whether all connected configurations of entities in exhaustive models of typical HPSG grammars can be regarded empirical configurations. My hypothesis is that all connected configurations in exhaustive models should be empirical, because the actual existence in the world or the well-formedness of non-empirical configurations cannot be verified by any means and should thus not be in the domain of an empirical science. My dominant concern at present is that exhaustive models of existing HPSG grammars actually do contain non-empirical entities of various kinds, because the grammars were never formulated with sufficient attention to their denotation. Exhaustive models give us a unique opportunity to investigate the denotation of grammars, to identify problematic aspects, and to solve the problems by eliminating problematic configurations. If current grammars are formulated too loosely, we can make their theories more rigorous and use the logic to formulate additional background assumptions which have so far been left implicit. This strategy should ultimately lead to satisfactory exhaustive models.

Once we have obtained satisfactory models, we can return to the philosophical issues of the ontology of models. The results obtained from investigating HPSG grammars and their linguistic content will shed new light on the (hidden) commitments involved in choosing from existing model theories for HPSG grammars. The insights we gain from considering linguistic theory alongside the logical architecture of the formalism and philosophical convictions should ultimately lead to a better understanding of what can be a satisfactory theory of the linguistic meaning of grammars in a constraint-based framework.

Before embarking on the investigation of the basic hypotheses of the HPSG framework about the structure of natural languages, the next section takes a closer look at the structure of relations in RSRL models. It will propose a minor modification of RSRL's notion of interpretations of signatures, which will lay to rest certain worries which linguists have aired about the effect of relation principles on grammar models. As we will demonstrate, the modification does not affect the theory of exhaustive models.

1.3 RSRL without Relational Monsters

In the previous sections I presented RSRL with a new syntax for its formal languages. The new syntax is better suited for applications of RSRL in the HPSG framework. As is shown in Chapter 3, the new version of RSRL is just a syntactic variant and does not change anything substantial. The innovation of the present section goes beyond this type of modification and addresses a question which some linguists have raised regarding certain structures in models of grammars. My goal here is to give a general characterization of these troublesome structures and to modify the notion of interpretations of signatures in such a way that these structures are eliminated without any further model-theoretic consequences for RSRL grammars.

When I introduced the concepts underlying the design of the logical languages of RSRL at the beginning of Section 1.1, I immediately stressed that the languages of RSRL were created to talk about connected configurations of entities. This idea is so much at the heart of the formalism that it is impossible even to make statements which relate in any way unconnected configurations in an interpretation. For example, it is impossible to make the shape of one configuration dependent on the shape of another configuration if one is not a subconfiguration of the other.

While this observation needs no qualification, there is one point in the design of RSRL which apparently does not pay attention to this fundamental fact about the formalism. This point is in the semantic structures. To be more precise, the configurations in question concern relations. In order to understand the issue it is best to take a close look at an example.

Figure 1.2 shows a model of the grammar $\langle \Sigma_1, \theta_2 \rangle$ with two connected configurations of entities, one for the word *Uther* and one for the word *walks*. I will call this $\langle \Sigma_1, \theta_2 \rangle$ model I_3 .

Figure 1.3 shows another $\langle \Sigma_1, \theta_2 \rangle$ model. It is very similar to I_3 and contains isomorphic configurations of *Uther* and *walks* as far as the two graphs in the picture are concerned. However, there is a small difference between the two $\langle \Sigma_1, \theta_2 \rangle$ models. In the second model, which I will henceforth call I_4 , there are more tuples in the two relations member and append than in I_3 . How can this be, considering that I_3 and I_4 are models of $\langle \Sigma_1, \theta_2 \rangle$, and $\langle \Sigma_1, \theta_2 \rangle$ comprises an APPEND PRINCIPLE and a MEMBER PRINCIPLE, which are supposed to fix the meaning of the relation symbols append and member to those tuples of entities intuitively expected to be in their denotation in $\langle \Sigma_1, \theta_2 \rangle$ models? How can there be any variation of append and member tuples in $\langle \Sigma_1, \theta_2 \rangle$ models if we can capture our intuitions about relations reasonably well with relation principles?

The reason for this surprising behavior of $\langle \Sigma_1, \theta_2 \rangle$ models is a property that all the tuples which distinguish member and append in I_3 and I_4 have in common: All the extra tuples in I_4 , $\langle 10, 4 \rangle$, $\langle 13, 5 \rangle$, $\langle 4, 9 \rangle$, $\langle 2, 2, 14 \rangle$, $\langle 2, 2, 13 \rangle$, and $\langle 0, 6, 9 \rangle$, contain nodes which occur in two configurations which are not connected by attribute arcs. For example the nodes numbered 10, 13 and 9 belong to *walks*, whereas the nodes numbered 4 and 5 belong to *Uther*. Each extra tuple in I_4 contains at least one entity from *walks* and at least one entity from *Uther*. On the other hand, it is not possible to add or to remove any tuple of entities from one and the same connected configuration in I_3 from the relations append and member without turning I_3 into an interpretation which is not a $\langle \Sigma_1, \theta_2 \rangle$ model. The APPEND PRINCIPLE and the MEMBER PRINCIPLE determine unique sets of append and member tuples with respect to entities from connected configurations, but they do not put restrictions on the to relations with respect to tuples which come from different connected configurations.

This fact about the effect of relation principles on their models is of course to be ex-



$$\begin{array}{l} \texttt{member} = \left\{ \begin{array}{l} \langle 2,1 \rangle, \langle 8,7 \rangle, \langle 12,11 \rangle \end{array} \right\} \\ \texttt{append} = \left\{ \begin{array}{l} \langle 1,3,1 \rangle, \langle 3,1,1 \rangle, \langle 3,3,3 \rangle, \langle 3,7,7 \rangle, \langle 3,11,11 \rangle, \langle 3,14,14 \rangle, \langle 7,14,7 \rangle, \langle 11,14,11 \rangle, \langle 14,1,1 \rangle, \\ \langle 14,3,3 \rangle, \langle 14,7,7 \rangle, \langle 14,11,11 \rangle, \langle 14,14,14 \rangle \end{array} \right\} \end{array}$$





$$\begin{array}{l} \texttt{member} = \left\{ \begin{array}{l} \langle 2,1\rangle, \langle 8,7\rangle, \langle 12,11\rangle, \langle 10,4\rangle, \langle 13,5\rangle, \langle 4,9\rangle \end{array} \right\} \\ \texttt{append} = \left\{ \begin{array}{l} \langle 1,3,1\rangle, \langle 3,1,1\rangle, \langle 3,3,3\rangle, \langle 3,7,7\rangle, \langle 3,11,11\rangle, \langle 3,14,14\rangle, \langle 7,14,7\rangle, \langle 11,14,11\rangle, \langle 14,1,1\rangle, \\ \langle 14,3,3\rangle, \langle 14,7,7\rangle, \langle 14,11,11\rangle, \langle 14,14,14\rangle, \langle 2,2,14\rangle, \langle 2,2,13\rangle, \langle 0,6,9\rangle \end{array} \right\} \end{array}$$

Figure 1.3: The $\langle \Sigma_1, \theta_2 \rangle$ model I_4

pected given the fact that the description languages of the formalism are not expressive enough to make properties of one connected configuration dependent upon properties of an independent second connected configuration. The (lack of) expressiveness of the logical languages implies that if it is possible for elements in the same relation tuple to belong to different connected configurations in an interpretation, the languages of RSRL cannot characterize this kind of relational structure. The reasons are easy to see: According to their syntactic structure, relational expressions must contain free variables, the variables in the argument position (DEFINITION 8, page 29). But according to the definition of grammars all expressions in grammar theories must be descriptions, which means that all variables occurring in principles in theories have to be bound by the existential or by the universal quantifier (DEFINITION 19, page 41). The meaning of the two quantifiers is defined in terms of quantification over components of entities (DEFINITION 14, page 34). Principles such as the APPEND PRINCIPLE and the MEMBER PRINCIPLE can, therefore, only make statements about tuples of entities which are all components of an entity. Relation principles cannot exclude entities which do not have at least one common topmost element in the universe from being in relational structures in models of grammars, nor can they force any of them to be in relational structures in models. The languages of RSRL simply cannot talk about these mixed tuples at all. It follows that mixed tuples may or may not occur in models, including exhaustive models, and that the grammar writer has no influence over this. There is no description which could distinguish between I_3 and I_4 . To give a name to the phenomenon at hand, let me call relations with tuples consisting of entities from different connected configurations in an interpretation *monster relations*.

The potential presence of monster relations in RSRL grammar models was noted very early on. The reason for their existence is not that they were not noticed at the time RSRL was created, but rather that leaving the possibility of more general relational configurations in interpretations in the formalism was mathematically more elegant and did not interfere with the main purpose and the proper functioning of the formalism. The earliest discussion of monster relations probably took place among Paul King, Kiril Simov and me in connection with work which aimed at characterizing model-theoretically the relationship between the expressive power of SRL and RSRL.¹¹ Around the same time Adam Przepiórkowski and Tilman Höhle (p.c.) independently noticed this property of RSRL relations as well and expressed some concern about it for purely linguistic reasons. As linguists who were thinking about how they could describe natural languages, they were worried about not having full control over the structure of relations in exhaustive grammar models. Moreover, they did not see any empirical significance of monster relations and concluded that they should not occur in the intended exhaustive models of linguistic grammars. The variant of RSRL to be presented below is motivated by such linguistic concerns and shows how to solve the problem.

In the early days of RSRL, however, when the study of mathematical properties took precedence over the study of models of particular grammars, it was decided that there was not enough reason for concern to justify any loss of elegance in definitions which

 $^{^{11} \}mathrm{Unfortunately}$ the resulting manuscript was never finished and remains unpublished.

might complicate the structure of the formalism, because the possible presence of monster relations in models of grammars did not influence the model-theoretic properties of RSRL which we were primarily interested in at the time. In particular, since exhaustive models can be defined on the basis of a notion of descriptive indiscernibility, and since monster relations are beyond the descriptive power of RSRL, their existence had no effect on the theory of exhaustive models, or on the relationship of exhaustive models of grammars to the abstract feature structure models in the spirit of [Pollard and Sag, 1994] and the type-oriented models of [Pollard, 1999].

In the face of linguists' concerns, and in the absence of any use for monster relations, I believe that monster relations should be eliminated by imposing stricter conditions on possible relational structures in interpretations. If it is possible to modify interpretations minimally in such a way that only monster relations are excluded but nothing else changes, it follows immediately that the exhaustive models of RSRL grammars remain unaltered in their relevant substance. King's theory of grammatical meaning is then still applicable, and the well studied relationship of exhaustive models to other structures which were used as explanations of grammatical meaning also remains intact.

In short, a change in the definition of interpretations which is limited to eliminating monster relations will exclude structures which caused uneasiness for at least some linguists, without changing any of those properties of RSRL which have been subject to modeltheoretic investigations. In the remainder of this section I will demonstrate how this can be achieved.

The solution proceeds in four steps. I will start with *small interpretations*, which are just like the original RSRL interpretations except that they do not contain a relation interpretation function. Small interpretations are sufficient to define the set of components of each entity in an interpretation, because componenthood is determined by accessibility by a term, and term interpretation only needs to refer to an identity function for the reserved colon symbol and to attribute interpretation functions. On the basis of componenthood in small interpretations it is possible to define the *set of possible relation tuples* as the set of tuples of entities which are components of at least one common predecessor. The set of possible relation tuples is then used to define full interpretations with a relation interpretation function which is appropriately restricted.

The definition of small interpretations is a simplified version of the original definition of interpretations, DEFINITION 3, page 21:

Definition 22 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, I is a small Σ interpretation iff

I is a triple (U, S, A),
U is a set,
S is a total function from U to S,
A is a total function from A to the set of partial functions from U to U,

for each $\alpha \in \mathcal{A}$ and each $u \in \mathsf{U}$,

if
$$A(\alpha)(u)$$
 is defined
then $\mathcal{F}\langle S(u), \alpha \rangle$ is defined, and $S(A(\alpha)(u)) \sqsubseteq \mathcal{F}\langle S(u), \alpha \rangle$, and

for each $\alpha \in \mathcal{A}$ and each $u \in \mathsf{U}$,

if
$$\mathcal{F}\langle \mathsf{S}(u), \alpha \rangle$$
 is defined then $\mathsf{A}(\alpha)(u)$ is defined.

Small interpretations are our original interpretations with the relation interpretation function omitted.

Small Σ interpretations are sufficient to define the set of components of an entity in a (small) interpretation. The necessary minor revision of DEFINITION 11, page 31, is illustrated below for the sake of completeness. DEFINITION 23 suffices as the definition of componenthood even after the introduction of full interpretations in DEFINITION 25, since our notion of componenthood is independent of relations.

Definition 23 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, for each small Σ interpretation $I = \langle U, S, A \rangle$, and for each $u \in U$,

$$\mathsf{Co}_{\mathsf{I}}^{u} = \left\{ u' \in \mathsf{U} \middle| \begin{array}{l} \textit{for some ass} \in Ass_{\mathsf{I}}, \\ \textit{for some } \pi \in \mathcal{A}^{*}, \\ T_{\mathsf{I}}^{ass}(:\pi)(u) \textit{ is defined, and} \\ u' = T_{\mathsf{I}}^{ass}(:\pi)(u) \end{array} \right\}.$$

Note that DEFINITION 23 presupposes the term interpretation functions $T_{\rm I}^{ass}$ with respect to small interpretations. Strictly speaking, we would have to insert here the relevant modification of the definition, but we will omit it. The omission is unproblematic, because DEFINITION 10 of term interpretation functions does not refer to the relation interpretation functions of full RSRL interpretations.

With all prerequisites in place, I can now capture the tuples of entities which belong to connected configurations:

Definition 24 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, and for each small Σ interpretation $I = \langle U, S, A \rangle$,

$$\mathsf{RT}_{\mathsf{I}} = \bigcup_{n \in \mathbb{N}} \left\{ \langle u_1, \dots, u_n \rangle \in \overline{\mathsf{U}}^n \middle| \begin{array}{l} \textit{for some } u \in \mathsf{U}, \\ \textit{for each } i \in \mathbb{N}, \ 1 \leq i \leq n, \\ u_i \in \overline{\mathsf{Co}_{\mathsf{I}}^u} \end{array} \right\}.$$

For each signature Σ and each (small) Σ interpretation I, I call RT₁ the set of possible relation tuples in I. Each possible relation tuple is a tuple with the characteristic that all entities in the tuple are components of a predecessor in the interpretation.¹² This

¹²Since relations might also have sequences of entities as arguments, the definition of possible relation tuples is constructed accordingly and allows each u_i to be a sequence of components.

means that all combinations of entities in connected configurations belong to a single possible relation tuple, and possible relation tuples must be composed of entities which are components of a common element. Alternatively, we can consider this from the descriptive point of view: For a given interpretation I, each tuple in the set RT_1 contains entities in configurations whose relationship can, in principle, be determined in descriptions, and no tuples are excluded whose elements can be determined by a description.

This is exactly the kind of restriction which linguists would like to put on relation structures. At the same time it is unproblematic for the theory of exhaustive models, because the restriction cannot be expressed within the logical languages of the formalism. We can immediately generalize the notion of possible relation tuples in a small interpretation to relation tuples in the redefined full interpretations of RSRL, where relation tuples will be drawn from the set of possible relation tuples from the corresponding small interpretations.

DEFINITION 25 replaces my former definition of interpretations of signatures, DEFINI-TION 3:

Definition 25 For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, and for each small Σ interpretation $I_s = \langle U, S, A \rangle$, I is a Σ interpretation iff

 I is a quadruple $\langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

R is a total function from \mathcal{R} to the power set of $\mathsf{RT}_{\mathsf{I}_s}$, and

for each
$$\rho \in \mathcal{R}$$
, $\mathsf{R}(\rho) \subseteq \left(\mathsf{RT}_{\mathsf{I}_s} \cap \overline{\mathsf{U}}^{\mathcal{AR}(\rho)}\right)$.

Relation symbols are still interpreted as tuples of entities (and sequences of entities) according to their arity, but these tuples are now restricted to the set of possible relation tuples in the corresponding small interpretation.

Returning to the two $\langle \Sigma_1, \theta_2 \rangle$ models I_3 and I_4 under the old definition of Σ interpretations, we can now inspect the consequences of the new definition of Σ interpretations for monster relations. Assume that we take the two graphs for *Uther* and *walks* as depicted in Figure 1.2, i.e., the depicted configurations of entities constitute the small Σ_1 interpretation whose influence on the corresponding full Σ_1 interpretation extensions we want to investigate. We find that I_3 is now the only possible $\langle \Sigma_1, \theta_2 \rangle$ model we can build from the given small Σ_1 interpretation. I_4 is no longer a $\langle \Sigma_1, \theta_2 \rangle$ model, because it is not even a Σ_1 interpretation any more: The tuples $\langle 10, 4 \rangle$, $\langle 13, 5 \rangle$, $\langle 4, 9 \rangle$, $\langle 2, 2, 14 \rangle$, $\langle 2, 2, 13 \rangle$, and $\langle 0, 6, 9 \rangle$ are not in the range of the relation interpretation function R according to DEFINITION 25 because they are not in the set of possible relation tuples in the small interpretation. The problem of monster relations has disappeared.

In all following chapters I will adopt the revised definition of interpretations without making a terminological distinction between the former and the new definitions. unembedded signs is acknowledged, but the theory is not spelled out, since it is not the real concern of constructional theories.

From our brief overview I conclude that important empirical properties of linguistic expressions in phonology, semantics, pragmatics and syntax are closely linked to utterances as the smallest units in which they can be observed and described in all relevant aspects. Units of language which are smaller than utterances become empirical by virtue of being components of utterances. Hence, HPSG grammars can be formulated more comprehensively and with greater accuracy if they include the concept of unembedded signs explicitly in their ontology.

2.1.3 Summary: A Grammar and an Intended Model

The preceding findings about the basic hypotheses in HPSG concerning the structure of natural languages and about the sign-based nature of the meaning of grammars can best be summarized in form of a small grammar and its intended model. For this purpose, I will repeat the grammar $\langle \Sigma_1, \theta_2 \rangle$ from Chapter 1 in a slightly different notation and provide a model for it. I will call this grammar $\langle \Sigma_t, \theta_t \rangle$, and I will refer to it throughout this chapter by its new name. The $\langle \Sigma_t, \theta_t \rangle$ model I will choose is supposed to reflect the expectations a linguist would usually have regarding the meaning of the grammar $\langle \Sigma_t, \theta_t \rangle$. The choice of the model is based on the assumption that linguists working in the HPSG framework regard unembedded signs as the central empirical entities of languages, and therefore expect to find connected configurations of entities under signs in the intended models of their grammars.

The grammar is consistent with the standard hypotheses about the structure of signs in Section 2.1.1, although it simplifies them considerably. In particular the SYNSEM attribute and the distinction between local and nonlocal properties is omitted, along with a number of optional grammar modules such as morphology and a worked out phonology. The remaining functions of the SYNSEM attribute are fulfilled by the attribute CAT.

The grammar $\langle \Sigma_t, \theta_t \rangle$ does not simply copy the principles of the grammar $\langle \Sigma_1, \theta_2 \rangle$, rather it presents them in a more reader friendly format. The syntax of the descriptions in the theory θ_t follows the simplifying notational conventions presented in Section 3.1. They make the notation much more compact and should be immediately transparent, even without consulting the explanations in Section 3.1.

The signature remains unchanged:

(13) The signature Σ_t :

```
top
   sign
          PHON
                  list
          CAT
                  cat
      phrase
               H_DTR
                         sign
               NH_DTR
                         sign
      word
   list
      nelist
              FIRST
                      top
              REST
                      list
      elist
   cat
         HEAD
                   head
         SUBCAT
                   list
   head
      verb
      noun
   phonstring
      uther
      walks
Relations
member/2
append/3
```

(14) enumerates the principles in the theory θ_t . Note that the member relation is not used in any of the substantive grammatical principles (14a)–(14d). The member relation and the MEMBER PRINCIPLE are still included in this grammar, as they typically belong to any larger HPSG grammar. Most importantly, the member relation is a much simpler example of a relation than the append relation. It will thus be useful in our discussion of the models of $\langle \Sigma_t, \theta_t \rangle$.

(14) a. WORD PRINCIPLE:

$$\left(\begin{bmatrix} \text{PHON } \langle uther \rangle \\ \text{CAT } \begin{bmatrix} \text{HEAD } noun \\ \text{SUBCAT } elist \end{bmatrix} \right) \lor \begin{bmatrix} \text{PHON } \langle walks \rangle \\ \text{CAT } \begin{bmatrix} \text{HEAD } verb \\ \text{SUBCAT } \langle \begin{bmatrix} \text{HEAD } noun \\ \text{SUBCAT } elist \end{pmatrix} \end{bmatrix} \right)$$

b. ID PRINCIPLE:

 $[phrase] \rightarrow \begin{bmatrix} CAT \ SUBCAT \ elist \\ H-DTR \ CAT \ SUBCAT \ & \langle \blacksquare \rangle \\ NH-DTR \ CAT \ \blacksquare \end{bmatrix}$

c. HEAD FEATURE PRINCIPLE: $\begin{bmatrix} phrase \end{bmatrix} \rightarrow \begin{bmatrix} CAT \text{ HEAD } \fbox{} \\ H_DTR \text{ CAT HEAD } \fbox{} \end{bmatrix}$ d. Constituent Order Principle:

$$[phrase] \rightarrow \left(\begin{bmatrix} PHON & \boxed{3} \\ H_DTR & PHON & \boxed{2} \\ NH_DTR & PHON & \boxed{1} \end{bmatrix} \land \texttt{append}(\boxed{1}, \boxed{2}, \boxed{3}) \right)$$

e. Append Principle:

$$\forall \texttt{I} \forall \texttt{I} \forall \texttt{I} \forall \texttt{I} \\ \left(\texttt{append}(\texttt{I},\texttt{2},\texttt{3}) \leftrightarrow \left(\begin{array}{c} (\texttt{I}[\textit{elist}] \land \texttt{2}[\textit{list}] \land \texttt{2} = \texttt{3}) \lor \\ \exists \texttt{4} \exists \texttt{5} \exists \texttt{6} \\ (\land \texttt{4} \mid \texttt{5}) \land \texttt{3} \langle \texttt{4} \mid \texttt{6} \rangle \\ \land \texttt{append}(\texttt{5},\texttt{2},\texttt{6}) \end{array} \right) \end{array} \right) \right)$$

f. Member Principle:

$$\forall \texttt{I}\forall \texttt{2} \left(\texttt{member}(\texttt{I},\texttt{2}) \leftrightarrow \left(\begin{array}{c} \texttt{2} \langle \texttt{1} \mid \textit{list} \rangle \lor \\ \exists \texttt{3} \left(\texttt{2} \langle \textit{top} \mid \texttt{3} \rangle \land \texttt{member}(\texttt{1},\texttt{3}) \right) \end{array} \right) \right)$$

We may assume that the utterances licensed by our grammar are the phrase *Uther walks*, and the words *Uther* and *walks*. For the sentence *Uther walks* this assumption is certainly uncontroversial. To see its plausibility for the word *Uther*, consider the exclamation *Uther!*, which can easily be construed as a complete utterance in contexts of wonder or rebuke.

It is more difficult to construct an appropriate context for an independent use of the word *walks*. How convincing the case for its existence is depends on decisions about the overall syntactic architecture of grammar. I will not try to provide a compelling argument here, but simply postulate that at least in some restricted contexts, possibly to be linked to the theory of elliptical constructions, the grammar of English licenses utterances which consist only of the word *walks*. Under this assumption the word *walks* should occur as an independent utterance with illocutionary force in models of English grammars.

Can we find the three signs with the phonologies Uther, walks and Uther walks in exhaustive models of $\langle \Sigma_t, \theta_t \rangle$? Yes, we can: Figure 2.1 shows a (non-exhaustive) model, I₃, of our grammar with the sentence Uther walks and the separate words Uther and walks. By definition we know that all exhaustive models of $\langle \Sigma_t, \theta_t \rangle$ must contain at least one isomorphic copy of each connected configuration of entities depicted in Figure 2.1.

Two properties of the model I_3 can be taken as first indications that there are problems with the grammar $\langle \Sigma_t, \theta_t \rangle$ and its intended meaning in terms of the class of exhaustive $\langle \Sigma_t, \theta_t \rangle$ models.

The first shortcoming has to do with an apparent weakness of the requirement that an exhaustive model contain at least one isomorphic copy of each configuration of entities. In a sense, this requirement does not seem to be strong enough to guarantee the intended meaning of $\langle \Sigma_t, \theta_t \rangle$. To see this, note that the configurations of *Uther* (under entity 30) and *walks* (under entity 19) can have isomorphic counterparts in models with fewer entities than I_3 , even if the model in question also comprises a configuration of the sentence *Uther walks*. The reason is this: A model with a single connected configuration isomorphic to *Uther walks* (under entity 16) suffices to fulfill the condition with respect to the three signs *walks*, *Uther and Uther walks*, since a model of this kind already comprises isomorphic copies of *Uther and walks*. In the model I_3 the relevant isomorphic copies of *Uther* and



$$\begin{split} \mathtt{member} &= \left\{ \begin{array}{l} \langle 2,17 \rangle, \langle 2,1 \rangle, \langle 4,12 \rangle, \langle 9,17 \rangle, \langle 9,8 \rangle, \langle 32,31 \rangle, \langle 29,27 \rangle, \langle 23,21 \rangle \end{array} \right\} \\ & \left\{ \begin{array}{l} \langle 1,3,1 \rangle, \langle 1,8,17 \rangle, \langle 3,1,1 \rangle, \langle 3,3,3 \rangle, \langle 3,6,6 \rangle, \langle 3,8,8 \rangle, \langle 3,12,12 \rangle, \langle 3,13,13 \rangle, \\ \langle 3,14,14 \rangle, \langle 3,15,15 \rangle, \langle 3,17,17 \rangle, \langle 6,1,1 \rangle, \langle 6,3,3 \rangle, \langle 6,6,6 \rangle, \langle 6,8,8 \rangle, \langle 6,12,12 \rangle, \\ \langle 6,13,13 \rangle, \langle 6,14,14 \rangle, \langle 6,15,15 \rangle, \langle 6,17,17 \rangle, \langle 8,15,8 \rangle, \langle 12,14,12 \rangle, \langle 13,1,1 \rangle, \\ \langle 13,3,3 \rangle, \langle 13,6,6 \rangle, \langle 13,8,8 \rangle, \langle 13,12,12 \rangle, \langle 13,13,13 \rangle, \langle 13,14,14 \rangle, \langle 13,15,15 \rangle, \\ \langle 13,17,17 \rangle, \langle 14,1,1 \rangle, \langle 14,3,3 \rangle, \langle 14,6,6 \rangle, \langle 14,8,8 \rangle, \langle 14,12,12 \rangle, \langle 14,13,13 \rangle, \\ \langle 14,14,14 \rangle, \langle 14,15,15 \rangle, \langle 14,17,17 \rangle, \langle 15,1,1 \rangle, \langle 15,3,3 \rangle, \langle 15,6,6 \rangle, \langle 15,8,8 \rangle, \\ \langle 15,12,12 \rangle, \langle 15,13,13 \rangle, \langle 15,14,14 \rangle, \langle 12,15,15 \rangle, \langle 15,17,17 \rangle, \langle 17,15,17 \rangle, \langle 31,33,31 \rangle, \\ \langle 22,21,21 \rangle, \langle 22,22,22 \rangle, \langle 22,24,24 \rangle, \langle 22,27,27 \rangle, \langle 22,28,28 \rangle, \langle 24,21,21 \rangle, \langle 24,22,22 \rangle, \\ \langle 24,24,24 \rangle, \langle 24,27,27 \rangle, \langle 24,28,28 \rangle, \langle 27,28,27 \rangle, \langle 28,21,21 \rangle, \langle 28,22,22 \rangle, \langle 28,24,24 \rangle, \\ \langle 28,27,27 \rangle, \langle 28,28,28 \rangle \end{split} \right\} \end{split}$$

Figure 2.1: A $\langle \Sigma_t, \theta_t \rangle$ model, I_3 , with the words Uther, walks and the phrase Uther walks

walks can be found respectively under the entities 0 and 7 of the configuration Uther walks. If we want to find everything that the grammar is intended to describe as an independent connected configuration in the relevant models, something needs to be changed. To require the presence of at least one isomorphic copy of each possible configuration is not enough.

The second problem is not immediately obvious, but the first signs of it become apparent by considering the surprisingly large number of tuples to be seen in the relation append. It is worth the trouble to spend some time investigating the relation append (and member) in I_3 . The relation tuples are, of course, determined by the MEMBER PRINCIPLE and the APPEND PRINCIPLE of the grammar. Whereas the interpretation of member is easy to reconcile with intuitions about the meaning of a membership relation it takes longer to see why the triples in append conform to the standard version of an append specification provided by the APPEND PRINCIPLE, however the reader is encouraged to check that each tuple shown in I₃ necessarily belongs to the given relations in $\langle \Sigma_t, \theta_t \rangle$ models.¹⁶ Close inspection reveals that the apparently artificially high number of triples in append is due to the encoding of lists in terms of entities in interpretations. In particular the multiple elist entities in each connected configuration contribute a large number of combinatorial possibilities, which do not correspond to immediate intuitions about an append relation in the given signs. When we investigate the full range of the possible models of $\langle \Sigma_t, \theta_t \rangle$, we will see that HPSG's list encoding leads to further complications. Our solution to these will also address the somewhat counterintuitive, albeit technically expected, behavior of append in I_3 .

After this brief introduction to the most salient problems with our HPSG grammar and its sign-based models, we will now turn to a systematic investigation of the form and models of HPSG grammars, taking $\langle \Sigma_t, \theta_t \rangle$ as an exemplary HPSG grammar whose size fits our purpose very well.

2.2 Toward Normal Form Grammars

The grammar $\langle \Sigma_t, \theta_t \rangle$ is consistent with the structural hypotheses about language of the HPSG framework; however it reduces the empirical coverage drastically compared to normal HPSG fragments of natural languages by omitting many details of more realistic grammars. This makes the number and size of connected configurations of entities described small enough to be inspected comfortably. At the same time models of this toy grammar have a sufficiently rich structure to illustrate my worries about models and exhaustive models in the denotation of typical HPSG grammars.

What do linguists who write a grammar such as $\langle \Sigma_t, \theta_t \rangle$ wish to find in the intended model of the grammar? Ignoring the ontological difference between types and possible tokens as the expected configurations in interpretations for the moment, it is fair to assume

¹⁶Note that models of the grammar would become even stranger with respect to the triples in **append** if I omitted the condition—usually disregarded by linguists—that the second and third argument of **append** always be lists. This condition is expressed by the second conjunct of the first disjunct (which is the 'base case') to the right of the bi-implication symbol of the APPEND PRINCIPLE (8e) on page 42.

Chapter 3

Technicalities

This chapter provides the technical background to the chapters on the mathematical and linguistic frameworks (Chapter 1 and 2) by presenting a few necessary mathematical details omitted there for expository reasons. Section 3.1 makes explicit a number of notational conventions which simplify the presentation of grammatical principles written in RSRL; Section 3.2 sketches a proof of my claim that the formalism defined in Chapter 1 is in fact a dialect of RSRL. The notational abbreviations in Section 3.1 were already used in the notation of the normal form grammar $\langle \Sigma_{Ex}, \theta_{Ex} \rangle$ (stated in (20) and (21)) in Section 2.3 and later in that chapter. They will be applied throughout Part II of this work.

3.1 Notational Conventions

The syntactic design of the formal languages of the new dialect of RSRL presented in Section 1.1 follows the notational conventions of the HPSG literature quite closely. However, as usual for mathematical systems, the AVM syntax of RSRL contains a number of technical notational devices designed to keep its recursive definition of syntax and the semantic interpretation as elegant and straightforward as possible. In linguistic principles, which are always AVM descriptions without free variables, a few simplifications can be introduced as notational conventions which make the formal languages significantly easier to use for linguists. In the present section I intend to loosen up the notation a bit in order to shrink unnecessarily large descriptions in grammars and to eliminate a few unwieldy technical details of the formal languages which are simply not necessary for the everyday purposes of linguists who use RSRL. The effect of the following conventions can best be observed by comparing the notation of the grammatical principles in the grammar $\langle \Sigma_1, \theta_1 \rangle$ in (8) on page 41 to the corresponding principles in the normal form grammar developed from the first grammar in (21) on page 113.

In order to simplify the notation of grammatical principles I will introduce a small number of notational conventions which will have a major impact on the size of many AVM descriptions. Most of the following conventions are cited from [Richter, 2004a, Chapter 3.2.3]; or, in the case of CONVENTIONS 5–7, they are modeled after the notation of the indirectly interpreted syntax of AVM formulae in [Richter, 2004a]. The conventions are designed to simplify the syntax of grammatical principles, and the underlying assumption is that the abbreviated expressions are all Σ AVM descriptions, that is, elements of \mathbb{AVM}_0^{Σ} .

First of all, whenever it can be done without causing confusion about the structure of the expression, I will omit the round brackets of AVM formulae which are conjoined by the standard logical connectives. In doing so, I follow the usual conventions of first order predicate logic about the operator precedence among the logical connectives. I have, in fact, already tacitly adopted this bracketing convention in the examples of grammatical principles in Section 1.1 above. In addition, I will also omit square brackets inside of Σ boxes whenever this can simplify the notation without becoming confusing. In particular, I will usually not write square brackets if a Σ box only contains a sort but no attributes, or if there is only one attribute but no sort mentioned (which is made possible by CONVENTION 3 below).

The remaining conventions are worth a more distinguished treatment and will be enumerated as conventions. In this form they can easily be consulted if the reader should find this necessary for checking the precise meaning of grammatical principles.

Clearly the pseudo variable colon can be omitted in front of top matrices if we agree that a maximal matrix without a tag is meant to be tagged by colon:¹

Convention 2 [Colon Convention] The colon as the tag of a maximal matrix may be left out in AVM descriptions.

Neither shall it be necessary to write an uninformative sort description in a Σ box only to ensure syntactic well-formedness:

Convention 3 [Metatop Convention] At the top of matrices, the symbol from the expanded sort set may be left out. If it is missing then I regard metatop as the missing symbol.

In grammatical principles of typical HPSG grammars, a significant number of quantifiers can be left out by adopting a simple quantificational closure convention for binding off variables which would otherwise remain free variables:

Convention 4 [Existential Closure Convention] If a variable is not explicitly bound by a quantifier, then I assume that it is captured by an implicit existential closure with wide scope over the entire formula.

The *Existential Closure Convention* should be used with care, in particular since all existential quantifiers can be left implicit. It is still necessary to explicitly mention all those existential quantifiers which occur in the scope of negation or universal quantifiers. This concerns, for example, all existential quantifiers in the principles which fix the meaning of relation symbols, such as the APPEND PRINCIPLE or the MEMBER PRINCIPLE.

Given the conventions above I can simplify the notation of the HEAD FEATURE PRIN-CIPLE of the small grammar in (8) considerably, writing it as follows:

 $^{^{1}}$ Recall that matrices which are not embedded in another matrix are called *maximal*.

(25)
$$phrase \rightarrow \begin{bmatrix} CAT HEAD \\ H_DTR CAT HEAD \end{bmatrix}$$

By our conventions, this stands for the description in (26), which can easily be shown to be equivalent to the original formulation:

 $(26) \exists \square \left(: [phrase] \rightarrow : \begin{bmatrix} metatop \\ CAT & [metatop \\ HEAD \square [metatop] \end{bmatrix} \\ H_DTR & [metatop \\ CAT & [metatop \\ HEAD \square [metatop]] \end{bmatrix} \end{bmatrix} \right)$

The next three conventions concern the description of lists and chains. Their purpose is to add the usual list notation with angled brackets to the syntax of AVM formulae, because it is much more readable than a notation with attribute symbols and sort symbols due to the optical distinction it makes between the description of lists and the description of other kinds of structures. Since it is not clear in advance which sort symbols and attribute symbols with their requisite appropriateness specifications are used to describe lists in a particular grammar, I first have to state in general terms the geometry of lists in signatures.² I will then define the list notation for any set of attributes and sorts which match this geometry. A similar notation will be introduced for chains. It will be followed by a convenient notation which will allow grammar writers to ignore the distinction between lists and chains by describing them simultaneously.

A signature Σ' is a fragment of a signature Σ if and only if each element in the septuple Σ' is a subset of the corresponding element of Σ . A list fragment of a signature Σ is a fragment Σ' of Σ such that it contains two attributes α_1 and α_2 , and at most four sorts: two species of Σ , σ_1 and σ_2 , together with their unique common immediate supersort, σ , and possibly one more sort of Σ , σ_{ε} .³ σ_1 and σ_2 are the only immediate subsorts of σ in Σ . In both Σ and Σ' the attributes α_1 and α_2 are appropriate to σ_1 but not to σ_2 and σ ; $\mathcal{F}\langle\sigma_1,\alpha_1\rangle = \sigma_{\varepsilon}$ and $\mathcal{F}\langle\sigma_1,\alpha_2\rangle = \sigma$. The set of relations is empty in Σ' .

I will call a signature which contains a list fragment a *signature with lists*. For signatures with lists I introduce a notational convention:

Convention 5 [List Convention] For each signature with lists Σ , assume that Σ' is a distinguished list fragment of Σ with attributes α_1 and α_2 , a sort σ subsuming the species σ_1 and σ_2 , and the two attributes appropriate to σ_1 .

• $\langle \rangle$ is another way of writing $[\sigma_2]$.

²Typical choices in the English language HPSG literature are the attributes FIRST and REST together with the sort symbols *list, elist* and *nelist*; however, variations like the attributes HD (head) and TL (tail) and sorts *list, e_list* and *ne_list* also occur frequently.

 $^{{}^{3}\}sigma_{\varepsilon}$ may be one of the previous three sorts.

• For each $n \in \mathbb{N}$, for each i with $1 \leq i \leq n$, for each $\beta_i \in \mathbb{BOX}^{\Sigma}$,

$$\langle \beta_1, \dots, \beta_n \rangle \text{ is an abbreviation for the unique } \beta \in \mathbb{UBOX}^{\Sigma} \text{ such that} \\ \beta = \begin{bmatrix} \sigma_1 \\ \alpha_1 & \beta_1 \\ \\ \alpha_2 & \dots & \begin{bmatrix} \sigma_1 \\ \alpha_1 & \beta_n \\ \\ \alpha_2 & \sigma_2 \end{bmatrix} .$$

• For each $n \in \mathbb{N}$ with $n \geq 2$, for each i with $1 \leq i \leq n$, for each $\beta_i \in \mathbb{BOX}^{\Sigma}$,

$$\langle \beta_1, \dots, \beta_{n-1} | \beta_n \rangle \text{ is an abbreviation for the unique } \beta \in \mathbb{UBOX}^{\Sigma} \text{ such that}$$

$$\beta = \begin{bmatrix} \sigma_1 \\ \alpha_1 & \beta_1 \\ \\ \alpha_2 & \dots & \begin{bmatrix} \sigma_1 \\ \alpha_1 & \beta_{n-1} \\ \\ \alpha_2 & \beta_n \end{bmatrix}].$$

For chains I will appeal to a very similar notational convention. Since the symbols for describing chains are fixed I do not need any additional auxiliary terminological conventions like the ones for the *List Convention*:

Convention 6 [Chain Convention]

- || || is another way of writing [echain].
- For each $n \in \mathbb{N}$, for each i with $1 \leq i \leq n$, for each $\beta_i \in \mathbb{BOX}^{\Sigma}$,

$$\|\beta_1, \dots, \beta_n\| \text{ is an abbreviation for the unique } \beta \in \mathbb{UBOX}^{\Sigma} \text{ such that}$$
$$\beta = \begin{bmatrix} nechain \\ \dagger & \beta_1 \\ & & \\ &$$

• For each $n \in \mathbb{N}$ with $n \geq 2$, for each i with $1 \leq i \leq n$, for each $\beta_i \in \mathbb{BOX}^{\Sigma}$,

$$\|\beta_1, \dots, \beta_{n-1} | \beta_n \| \text{ is an abbreviation for the unique } \beta \in \mathbb{UBOX}^{\Sigma} \text{ such that}$$
$$\beta = \begin{bmatrix} nechain \\ \dagger & \beta_1 \\ & &$$

It will be convenient to have a notation for describing lists and chains simultaneously. I use the term *tape* to refer to list-like structures and their descriptions when I do not want to make a precise distinction between lists and chains. Tapes typically occur in arguments of relations.

Convention 7 [Tape Convention] Assume that Σ is a signature with lists and that we have adopted the LIST CONVENTION (Convention 5) for a distinguished list fragment of Σ . Let \square be a variable.

- $\square \ll \gg$ is an abbreviation for $\square \langle \rangle \lor \square [echain].$
- For each $n \in \mathbb{N}$, for each i with $1 \leq i \leq n$, for each $\beta_i \in \mathbb{BOX}^{\Sigma}$,

 $\blacksquare \ll \beta_1, \ldots, \beta_n \gg is \text{ an abbreviation for } \blacksquare \langle \beta_1, \ldots, \beta_n \rangle \lor \blacksquare \| \beta_1, \ldots, \beta_n \|.$

• For each $n \in \mathbb{N}$ with $n \geq 2$, for each i with $1 \leq i \leq n$, for each $\beta_i \in \mathbb{BOX}^{\Sigma}$,

$$\square \ll \beta_1, \dots, \beta_n \gg \text{ is an abbreviation for}$$
$$\square \langle \beta_1, \dots, \beta_{n-1} | \beta_n \rangle \vee \square \| \beta_1, \dots, \beta_{n-1} | \beta_n \|.$$

With the final notational convention I want to provide more freedom for the use of relational formulae. Instead of only admitting variables in the argument positions of relational expressions, tagged Σ boxes may be written in these positions for an immediate description of the arguments. Such a notation is common in the HPSG literature:

Convention 8 [Relation Argument Convention] A tagged box, ϕ , may be written into an argument slot of a relational AVM formula. This notation is interpreted as abbreviating a conjunction of ϕ with the relational AVM formula which contains only the tag of ϕ in the respective argument slot. If the tag of ϕ is the colon, the notation is interpreted as described, except that the relevant argument slot of the relational AVM formula now contains the alphabetically first variable, v, which does not yet occur in the overall description, and the AVM equation := v is conjoined with the relational AVM formula and ϕ .

Consider the signature Σ_1 . With the *Relation Argument Convention* we can refer to verbs which have a saturated verbal argument on their SUBCAT list in the following compact way:

Without CONVENTION 8, this would have to be written in a slightly more complicated and much less perspicuous way:

(28)
$$\begin{bmatrix} word \\ \\ \\ CAT \begin{bmatrix} HEAD & verb \\ \\ SUBCAT \end{bmatrix}$$
 $\land 2 \begin{bmatrix} HEAD & verb \\ \\ \\ SUBCAT \\ \\ \\ \end{pmatrix}$ $\land member(2, 1)$

٦

It is immediately evident that the RELATION ARGUMENT CONVENTION becomes very effective when several relational expressions with descriptions of their arguments occur in a single description.

The RELATION ARGUMENT CONVENTION concludes the notational simplifications for the specification of grammatical principles.

3.2 Equivalence Proof

In this section, I will sketch a proof which shows that all grammars which are written in the syntax of Section 1.1 are indeed RSRL grammars, although they are written in a syntax which differs from the syntax of the original version of the RSRL formalism. In order to prove that the formalism in Chapter 1 is indeed the RSRL formalism from [Richter, 2004a] I have to show that for each RSRL grammar there is a grammar of the new version of the formalism such that an interpretation of their shared signature is a model of the RSRL grammar if and only if it is also a model of the grammar of the new formalism. And for each grammar of the new formalism there must be a corresponding grammar of the original RSRL formalism such that a model of the first is a model of the second grammar, and nothing which is not a model of one is a model of the other grammar. If we can find a pair of corresponding grammars in the two versions of the formalism with regard to their models no matter with which grammar and which formalism we start, we know that the two formalisms are the same.

For reasons of space I shall not repeat the complete definitions of the original RSRL formalism. I will adopt the syntax and semantics of RSRL presented in [Richter, 2004a, pp. 162–169]. In addition I will exploit other constructions from the same source which are convenient for my present purpose. For grammar writing an alternative syntax for RSRL is given in [Richter, 2004a] which is not directly interpreted. Instead, expressions in this syntax are translated into RSRL formulae by a translation function. The meaning of the translation of an expression is then taken as the meaning of the original expression. The uninterpreted syntax is very similar to the one which I used as the primary syntax in Section 1.1. For this reason I can also use the translation function in my proof with a few minor revisions which will be duly indicated.

Since we will show the relationship between two similar kinds of syntax and their meaning relations in order to demonstrate that they constitute the same formalism, we will have to be careful about the terminology lest we get confused about which syntax and meaning relation we are dealing with at any given point in the discussion. For this reason I will, for the remainder of this section, strictly adhere to the following nomenclature. The expressions of the original RSRL syntax will consistently be called (Σ) formulae and (Σ) descriptions. They must be distinguished from the (Σ) AVM formulae and (Σ) AVM descriptions of Section 1.1. Sets of descriptions will be called theories, and they will be

г

Bibliography

- [Asudeh and Crouch, 2002] Asudeh, Ash and Crouch, Richard 2002. Glue semantics for HPSG. In Dorothee Beermann Frank van Eynde, Lars Hellan (ed), Proceedings of the 8th International Conference on Head-Driven Phrase Structure Grammar, 1–19. CSLI Publications.
- [Barwise and Perry, 1983] Barwise, Jon and Perry, John 1983. *Situations and Attitudes*. Cambridge, Mass.: The MIT Press.
- [Bouma, 1994] Bouma, Gosse 1994. Calculated flexibility. In Harry Bunt, Reinhart Muskens, and Gerrit Rentier (eds), Proceedings of the International Workshop on Computational Semantics, 32–40. Katholieke Universiteit Brabant.
- [Bouma, 2003] Bouma, Gosse 2003. Verb clusters and the scope of adjuncts in Dutch. In Pieter A. M. Seuren and Gerard Kempen (eds), Verb Constructions in German and Dutch, 5–42. Amsterdam and Philadelphia: Benjamins.
- [Carnap, 1956] Carnap, Rudolf 1956. *Meaning and Necessity*. The University of Chicago Press, Chicago.
- [Carpenter, 1992] Carpenter, Bob 1992. *The Logic of Typed Feature Structures*. Cambridge University Press. Cambridge, Massachusetts, USA.
- [Chomsky, 1965] Chomsky, Noam 1965. Aspects of the Theory of Syntax. The M.I.T. Press.
- [Cooper, 1983] Cooper, Robin 1983. *Quantification and Syntactic Theory*. D. Reidel Publishing Company, Dordrecht.
- [Cooper, 1975] Cooper, Robin H. 1975. Montague's Semantic Theory and Transformational Syntax. PhD thesis, University of Massachusetts at Amherst.
- [Copestake, 2002] Copestake, Ann 2002. Implementing Typed Feature Structure Grammars, (= CSLI Lecture Notes, 110). Stanford: CSLI Publications.
- [Copestake et al., 2003] Copestake, Ann, Flickinger, Dan, Pollard, Carl, and Sag, Ivan A. 2003. Minimal Recursion Semantics: An introduction. *Note:* Journal submission, November 2003.

- [Corblin and Tovena, 2001] Corblin, Francis and Tovena, Lucia M. 2001. On the multiple expression of negation in Romance. In Yves D'Hulst, Johan Rooryck, and Jan Schroten (eds), Romance Languages and Linguistic Theory 1999, 87–115. John Benjamins, Amsterdam.
- [Curry, 1961] Curry, Haskell B. 1961. Some logical aspects of grammatical structure. In Jacobson (ed), Structure of Language and its Mathematical Aspects: Proceedings of the Twelfth Symposium in Applied Mathematics, 56–68. American Mathematical Society.
- [Dalrymple, 1999] Dalrymple, Mary (ed) 1999. Semantics and syntax in lexical functional grammar: the resource logic approach. MIT Press.
- [Daniels and Meurers, 2002] Daniels, Mike and Meurers, W. Detmar 2002. Improving the efficiency of parsing with discontinuous constituents. In Shuly Wintner (ed), Proceedings of NLULP'02: The 7th International Workshop on Natural Language Understanding and Logic Programming, (= Datalogiske Skrifter, 92), 49–68, Copenhagen: Roskilde Universitetscenter.
- [Davidson, 1980] Davidson, Donald 1980. Essays on Actions and Events. Clarendon Press, Oxford.
- [Dowty et al., 1981] Dowty, David R., Wall, Robert E., and Peters, Stanley 1981. Introduction to Montague Semantics. D. Reidel Publishing Company.
- [Ebert, 2003] Ebert, Christian 2003. On the expressive completeness of underspecified representations. In *Proceedings of the 14th Amsterdam Colloquium*, Amsterdam.
- [Egg, 1998] Egg, Markus 1998. Wh-questions in Underspecified Minimal Recursion Semantics. Journal of Semantics, 15:37–82.
- [Egg et al., 2001] Egg, Markus, Koller, Alexander, and Niehren, Joachim 2001. The Constraint Language for Lambda Structures. *Journal of Logic, Language and Information*, 10.4:457–485.
- [Frank and Reyle, 1995] Frank, Anette and Reyle, Uwe 1995. Principle based semantics for HPSG. In Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics, 9–16. Association for Computational Linguistics.
- [Gallin, 1975] Gallin, Daniel 1975. Intensional and Higher-Order Modal Logic. North-Holland, Amsterdam.
- [Gazdar et al., 1985] Gazdar, Gerald, Klein, Ewan, Pullum, Geoffrey K., and Sag, Ivan A. 1985. Generalized Phrase Structure Grammar. Harvard University Press. Cambridge Massachusetts.

- [Ginzburg and Sag, 2000] Ginzburg, Jonathan and Sag, Ivan A. 2000. Interrogative Investigations. The Form, Meaning, and Use of English Interrogatives. CSLI Publications.
- [Groenendijk and Stokhof, 1982] Groenendijk, Jeroen and Stokhof, Martin 1982. Semantic analysis of *wh*-complements. *Linguistics and Philosophy*, 5:175–233.
- [Haegeman and Zanuttini, 1991] Haegeman, Liliane and Zanuttini, Raffaella 1991. Negative Heads and the Neg Criterion. The Linguistic Review, 8:233–251.
- [Halvorsen and Ladusaw, 1979] Halvorsen, Per-Kristian and Ladusaw, William A. 1979. Montague's 'Universal Grammar': An introduction for the linguist. *Linguistics and Philosophy*, 3:185–223.
- [Hamm, 1999] Hamm, Fritz 1999. Modelltheoretische Untersuchungen zur Semantik von Nominalisierungen. SfS-Report-01-00, Seminar für Sprachwissenschaft der Universität Tübingen. Note: Habilitationsschrift.
- [Hendriks, 1993] Hendriks, Herman 1993. Studied Flexibility, (= ILLC Dissertation Series 1995-5). Institute for Logic, Language and Computation, Amsterdam.
- [Hindley and Seldin, 1986] Hindley, J. Roger and Seldin, Jonathan P. 1986. Introduction to Combinators and the lambda-Calculus. Cambridge University Press.
- [Höhle, 1999] Höhle, Tilman N. 1999. An architecture for phonology. In Robert D. Borsley and Adam Przepiórkowski (eds), *Slavic in Head-Driven Phrase Structure Grammar*, 61– 90. Stanford: CSLI Publications.
- [Janssen, 1983] Janssen, Theodoor Maria Victor 1983. Foundations and Applications of Montague Grammar. PhD thesis, University of Amsterdam.
- [Janssen, 1997] Janssen, Theo M. V. 1997. Compositionality. In Johan van Benthem and Alice ter Meulen (eds), Handbook of Logic and Language, 417–473. Elsevier Science B.V.
- [Johnson, 1988] Johnson, Mark 1988. Attribute-Value Logic and the Theory of Grammar. CSLI Publications.
- [Kasper, 1996] Kasper, Robert T. 1996. Semantics of recursive modification. Note: Unpublished Manuscript, September 11th, 1996. The Ohio State University.
- [Kathol, 1995] Kathol, Andreas 1995. *Linearization-Based German Syntax*. PhD thesis, Ohio State University.
- [Kathol, 2000] Kathol, Andreas 2000. Linear Syntax. Oxford University Press.
- [Kathol and Pollard, 1995] Kathol, Andreas and Pollard, Carl 1995. Extraposition via complex domain formation. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, 174–180.

- [Kepser, 2004] Kepser, Stephan 2004. On the complexity of RSRL. In Lawrence S. Moss and Richard T. Oehrle (eds), *Electronic Notes in Theoretical Computer Science*. Elsevier.
- [Kepser and Mönnich, 2003] Kepser, Stephan and Mönnich, Uwe 2003. Graph properties of HPSG feature structures. In Gerhard Jäger, Paola Monachesi, Gerald Penn, and Shuly Wintner (eds), Proceedings of Formal Grammar 2003, 115–124.
- [King, 1989] King, Paul J. 1989. A logical Formalism for Head-Driven Phrase Structure Grammar. PhD thesis, University of Manchester.
- [King, 1999] King, Paul J. 1999. Towards Truth in Head-driven Phrase Structure Grammar. In Valia Kordoni (ed), Tübingen Studies in Head-Driven Phrase Structure Grammar, (= Arbeitspapiere des SFB 340, Nr. 132, Volume 2), 301–352. Eberhard-Karls Universität Tübingen.
- [Klein and Sag, 1985] Klein, Ewan and Sag, Ivan A. 1985. Type-driven translation. Linguistics and Philosophy, 8:163–201.
- [Meurers, 2000] Meurers, Walt Detmar 2000. Lexical Generalizations in the Syntax of German Non-Finite Constructions. Phil. dissertation, Eberhard-Karls Universität Tübingen. Note: Published as: Arbeitspapiere des SFB 340, Nr. 145.
- [Meurers et al., 2002] Meurers, W. Detmar, Penn, Gerald, and Richter, Frank 2002. A web-based instructional platform for constraint-based grammar formalisms and parsing. In Dragomir Radev and Chris Brew (eds), *Effective Tools and Methodologies for Teaching NLP and CL*, 18–25, New Brunswick, NJ: The Association for Computational Linguistics. *Note:* Proceedings of the Workshop held at the 40th Annual Meeting of the Association for Computational Linguistics. 7.–12. July 2002. Philadelphia, PA.
- [Montague, 1974a] Montague, Richard 1974a. English as a formal language. In Richmond H. Thomason (ed), Formal Philosophy. Selected Papers of Richard Montague, 188–221. Yale University Press.
- [Montague, 1974b] Montague, Richard 1974b. The proper treatment of quantification in ordinary English. In Richmond H. Thomason (ed), Formal Philosophy. Selected Papers of Richard Montague, 247–270. Yale University Press.
- [Montague, 1974c] Montague, Richard 1974c. Universal grammar. In Richmond H. Thomason (ed), Formal Philosophy. Selected Papers of Richard Montague, 222–246. Yale University Press.
- [Moshier, 1988] Moshier, Michael Andrew 1988. Extensions to Unification Grammar for the Description of Programming Languages. PhD thesis, University of Michigan.
- [Müller, 1999] Müller, Stefan 1999. Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche, (= Linguistische Arbeiten, 394). Max Niemeyer Verlag, Tübingen.

- [Müller, 2004] Müller, Stefan 2004. Continuous or discontinuous constituents? A comparison between syntactic analyses for constituent order and their processing systems. *Research on Language and Computation*, 2:209–257.
- [Muskens, 2001] Muskens, Reinhard 2001. Talking about trees and truth-conditions. Journal of Logic, Language and Information, 10.4:417–455.
- [Nerbonne, 1992] Nerbonne, John 1992. Constraint-based semantics. In Paul Dekker and Martin Stokhof (eds), Proceedings of the Eighth Amsterdam Colloquium, 425–444. Institute for Logic, Language and Information.
- [Partee, 1975] Partee, Barbara 1975. Montague grammar and transformational grammar. Linguistic Inquiry, 6.2:203–300.
- [Partee, 1976] Partee, Barbara H. 1976. Montague Grammar. Academic Press, New York.
- [Partee, 1996] Partee, Barbara H. 1996. The development of formal semantics in linguistic theory. In Shalom Lappin (ed), *The Handbook of Contemporary Semantic Theory*, 11–38. Blackwell Publishers.
- [Partee, 2004a] Partee, Barbara H. 2004a. Compositionality. In Compositionality in Formal Semantics, 153–181. Blackwell Publishing Ltd.
- [Partee, 2004b] Partee, Barbara H. 2004b. Compositionality in Formal Semantics. Blackwell Publishing Ltd.
- [Partee and Hendriks, 1997] Partee, Barbara H. and Hendriks, Herman L. W. 1997. Montague grammar. In Johan van Benthem and Alice ter Meulen (eds), Handbook of Logic and Language, 5–91. Elsevier Science B.V.
- [Penka and von Stechow, 2001] Penka, Doris and von Stechow, Arnim 2001. Negative Indefinita unter Modalverben. In Reimar Müller and Marga Reis (eds), Modalität und Modalverben im Deutschen, (= Linguistische Berichte. Sonderheft 9), 263–286. Helmut Buske Verlag, Hamburg.
- [Penn, 1999a] Penn, Gerald 1999a. A Generalized-Domain-Based Approach to Serbo-Croation Second Position Clitic Placement. In Gosse Bouma, Erhard Hinrichs, Geert-Jan M. Kruijff, and Richard T. Oehrle (eds), Constraints and Resources in Natural Language Syntax and Semantics, 119–136. CSLI Publications.
- [Penn, 1999b] Penn, Gerald 1999b. Linearization and WH-Extraction in HPSG: Evidence from Serbo-Croatian. In Robert D. Borsley and Adam Przepiórkowski (eds), Slavic in Head-Driven Phrase Structure Grammar, 149–182. CSLI Publications.
- [Penn, 2004] Penn, Gerald 2004. Balancing clarity and efficiency in typed feature logic through delaying. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 240–247.

- [Penn and Haji-Abdolhosseini, 2003a] Penn, Gerald and Haji-Abdolhosseini, Mohammad 2003a. *ALE. The Attribute Logic Engine User's Guide with TRALE extensions. Note:* Version 4.0 Beta. With contributions by other authors.
- [Penn and Haji-Abdolhosseini, 2003b] Penn, Gerald and Haji-Abdolhosseini, Mohammad 2003b. Topological parsing. In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, April 12–17, 2003, Budapest, Hungary, 283–290.
- [Penn and Richter, 2004] Penn, Gerald and Richter, Frank 2004. Lexical Resource Semantics: From theory to implementation. In *The 11th International Conference on Head-driven Phrase Structure Grammar. Abstracts*, 10–14. University of Leuven. Centre for Computational Linguistics.
- [Pollard and Sag, 1987] Pollard, Carl and Sag, Ivan A. 1987. Information-Based Syntax and Semantics. Vol.1: Fundamentals. CSLI Lecture Notes 13.
- [Pollard and Sag, 1994] Pollard, Carl and Sag, Ivan A. 1994. Head-Driven Phrase Structure Grammar. University of Chicago Press.
- [Pollard, 1999] Pollard, Carl J. 1999. Strong generative capacity in HPSG. In Gert Webelhuth, Jean-Pierre Koenig, and Andreas Kathol (eds), *Lexical and Constructional Aspects* of Linguistic Explanation, 281–297. CSLI Publications.
- [Pollard and Yoo, 1998] Pollard, Carl J. and Yoo, Eun Jung 1998. A unified theory of scope for quantifiers and *wh*-phrases. *Journal of Linguistics*, 34:415–445.
- [Przepiórkowski, 1998] Przepiórkowski, Adam 1998. 'A unified theory of scope' revisited. Quantifier retrieval without spurious ambiguities. In Gosse Bouma, Geert-Jan M. Kruijff, and Richard T. Oehrle (eds), Proceedings of the FHCG-98, 14–16 August 1998, Saarbrücken, 185–195.
- [Pullum and Scholz, 2001] Pullum, Geoffrey K. and Scholz, Barbara C. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In Philippe de Groote, Glyn Morrill, and Christian Retoré (eds), Logical Aspects of Computational Linguistics: 4th International Conference, 17–43. Berlin: Springer-Verlag.
- [Reape, 1989] Reape, Mike 1989. A logical treatment of semi-free word order and bounded discontinuous constituency. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics*, 103–110.
- [Reape, 1990] Reape, Mike 1990. Getting things in order. In *Proceedings of the Symposium on Discontinuous Constituency*. Institute for Language Technology and Artificial Intelligence.
- [Reape, 1992] Reape, Mike 1992. A Formal Theory of Word Order: A Case Study of West Germanic. PhD thesis, University of Edinburgh.

- [Reape, 1994] Reape, Mike 1994. Domain Union and Word Order Variation in German. In John Nerbonne, Klaus Netter, and Carl Pollard (eds), German in Head-Driven Phrase Structure Grammar, 151–197. CSLI Publications.
- [Reinhard, 2001] Reinhard, Sabine 2001. Deverbale Komposita an der Morphologie-Syntax-Semantik-Schnittstelle: ein HPSG-Ansatz. Phil. dissertation, Eberhard-Karls Universität Tübingen.
- [Reyle, 1993] Reyle, Uwe 1993. Dealing with ambiguities by underspecification: Construction, representation and deduction. *Journal of Semantics*, 10.2:123–179.
- [Richter, 1997] Richter, Frank 1997. Die Satzstruktur des Deutschen und die Behandlung langer Abhängigkeiten in einer Linearisierungsgrammatik. Formale Grundlagen und Implementierung in einem HPSG-Fragment. In Erhard Hinrichs, Detmar Meurers, Frank Richter, Manfred Sailer, and Heike Winhart (eds), Ein HPSG-Fragment des Deutschen, Teil 1: Theorie, 13–187. Eberhard-Karls Universität Tübingen.
- [Richter, 2004a] Richter, Frank 2004a. A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar. Phil. dissertation (2000), Eberhard-Karls Universität Tübingen.
- [Richter, 2004b] Richter, Frank 2004b. A web-based course in grammar formalisms and parsing. *Note:* Electronic textbook distributed over the e-learning platform ILIAS at the Seminar für Sprachwissenschaft of the Eberhard-Karls Universität Tübingen.
- [Richter et al., 2002] Richter, Frank, Ovchinnikova, Ekaterina, Trawiński, Beata, and Meurers, W. Detmar 2002. Interactive graphical software for teaching the formal foundations of Head-Driven Phrase Structure Grammar. In Gerhard Jäger, Paola Monachesi, Gerald Penn, and Shuly Wintner (eds), Proceedings of Formal Grammar 2002, 137–148.
- [Richter and Sailer, 1999a] Richter, Frank and Sailer, Manfred 1999a. A lexicalist collocation analysis of sentential negation and negative concord in French. In Valia Kordoni (ed), Tübingen Studies in Head-driven Phrase Structure Grammar, 231–300. Eberhard-Karls Universität Tübingen.
- [Richter and Sailer, 1999b] Richter, Frank and Sailer, Manfred 1999b. LF conditions on expressions of Ty2: An HPSG analysis of negative concord in Polish. In Robert D. Borsley and Adam Przepiórkowski (eds), *Slavic in Head-Driven Phrase Structure Grammar*, 247–282. Stanford: CSLI Publications.
- [Richter and Sailer, 1999c] Richter, Frank and Sailer, Manfred 1999c. Underspecified semantics in HPSG. In Harry C. Blunt and Reinhard Muskens (eds), *Computing Meaning*, 95–112. Dordrecht: Kluwer Academic Publishers.
- [Richter and Sailer, 2001] Richter, Frank and Sailer, Manfred 2001. On the left periphery of German finite sentences. In W. Detmar Meurers and Tibor Kiss (eds), Constraint-Based Approaches to Germanic Syntax, 257–300. Stanford: CSLI Publications.

- [Richter and Sailer, 2003] Richter, Frank and Sailer, Manfred 2003. Cranberry words in formal grammar. In Claire Beyssade, Olivier Bonami, Patricia Cabredo Hofherr, and Francis Corblin (eds), *Empirical Issues in Formal Syntax and Semantics* 4, 155–171. Presses de l'Université de Paris-Sorbonne.
- [Richter and Sailer, 2004a] Richter, Frank and Sailer, Manfred 2004a. Basic concepts of Lexical Resource Semantics. In Arnold Beckmann and Norbert Preining (eds), ESSLLI 2003 – Course Material I, (= Collegium Logicum, 5), 87–143. Kurt Gödel Society Wien.
- [Richter and Sailer, 2004b] Richter, Frank and Sailer, Manfred 2004b. Polish negation and lexical resource semantics. In Lawrence S. Moss and Richard T. Oehrle (eds), *Electronic Notes in Theoretical Computer Science*. Elsevier.
- [Richter et al., 1999] Richter, Frank, Sailer, Manfred, and Penn, Gerald 1999. A formal interpretation of relations and quantification in HPSG. In Gosse Bouma, Erhard Hinrichs, Geert-Jan M. Kruijff, and Richard Oehrle (eds), *Constraints and Resources in Natural Language Syntax and Semantics*, 281–298. Stanford: CSLI Publications.
- [Sailer, 2003] Sailer, Manfred 2003. Combinatorial semantics and idiomatic expressions in Head-Driven Phrase Structure Grammar. Phil. Dissertation (2000). Arbeitspapiere des SFB 340. 161, Eberhard-Karls Universität Tübingen.
- [Sailer, 2004] Sailer, Manfred 2004. Propositional relative clauses in German. In The 11th International Conference on Head-driven Phrase Structure Grammar. Abstracts, 155–159. University of Leuven. Centre for Computational Linguistics.
- [Sailer and Richter, 2002a] Sailer, Manfred and Richter, Frank 2002a. Collocations and the representation of polarity. In Gábor Alberti, Kata Balogh, and Paul Dekker (eds), *Proceedings of the Seventh Symposium on Logic and Language*, 129–138, Pécs.
- [Sailer and Richter, 2002b] Sailer, Manfred and Richter, Frank 2002b. Not for love or money: Collocations! In Gerhard Jäger, Paola Monachesi, Gerald Penn, and Shuly Wintner (eds), Proceedings of Formal Grammar 2002, 149–160.
- [Smolka, 1992] Smolka, Gert 1992. Feature-constraint logics for unification grammars. Journal of Logic Programming, 12:51–87.
- [Soehn, 2004] Soehn, Jan-Philipp 2004. Über Bärendienste und erstaunte Bauklötze. Idiome ohne freie Lesart in der HPSG. PhD thesis, Friedrich-Schiller-Universität Jena. Note: In preparation. Version of July 27th, 2004.
- [Soehn and Sailer, 2003] Soehn, Jan-Philipp and Sailer, Manfred 2003. At first blush on tenterhooks. About selectional restrictions imposed by nonheads. In Gerhard Jäger, Paola Monachesi, Gerald Penn, and Shuly Wintner (eds), Proceedings of Formal Grammar 2003, 149–161.

- [Suhre, 2000] Suhre, Oliver 2000. Computational aspects of a grammar formalism for languages with freer word order. Diplomarbeit, Eberhard-Karls Universität Tübingen. *Note:* Published in Arbeitspapiere des SFB 340, Nr. 154.
- [Thomason, 1974] Thomason, Richmond H. (ed) 1974. Formal Philosophy. Selected Papers of Richard Montague. Yale University Press.
- [Trawiński et al., 2004] Trawiński, Beata, Sailer, Manfred, and Soehn, Jan-Philipp 2004. Combinatorial aspects of collocational prepositional phrases. In Patrick Sain-Dizier (ed), *Computational Linguistics Dimensions of Syntax and Semantics of Prepositions*. Kluwer Academic Press. *Note:* To appear.
- [van Deemter and Peters, 1996] van Deemter, Kees and Peters, Stanley (eds) 1996. Semantic Ambiguity and Underspecification. CSLI Publications.
- [von Stechow, 1991] Stechow, Arnim von 1991. Theorie der Satzsemantik. In Arnin von Stechow and Dieter Wunderlich (eds), *Semantik: Ein internationales Handbuch der zeit*genössischen Forschung, 90–148. Walter de Gruyter.
- [von Stechow, 1993] Stechow, Arnim von 1993. Die Aufgaben der Syntax. In Joachim Jacobs, Arnim von Stechow, Wolfgang Sternefeld, and Theo Vennemann (eds), Syntax. Ein internationales Handbuch zeitgenössischer Forschung, 1–88. Walter de Gruyter.
- [Zimmermann, 1989] Zimmermann, Thomas Ede 1989. Intensional logic and two-sorted type theory. *The Journal of Symbolic Logic*, 54:65–77.