Seminar für Sprachwissenschaft
Abteilung Computerlinguistk
Universität Tübingen

SS 2002
July 5th, 2002

Grammar Formalisms and Parsing

Prof. Dr. Erhard W. Hinrichs
Frank Richter
Beata Trawiński

We are now ready for completing our syntax and semantics of HPSG grammars. Equipped with a syntax and semantics that comprise variables and quantification, the necessary syntactic extension of our language is trivial. Extending our feature structure semantics in a way appropriate for the intended usage of relational expressions in grammatical principles of the HPSG literature is more difficult. To obtain an adequate semantics, we will have to proceed from simple abstract feature structures under a given signature to relational abstract feature structures under a given signature. We will then focus on how to write principles of grammar that restrict the meaning of relation symbols like `member` and `append` to their intended interpretation in feature structure models of grammars, and on how relational expressions like these can be used in principles of grammar.

A few initial examples of important grammatical principles with relations of the HPSG grammar of English of Pollard and Sag will give us an idea of what relations are meant to do in the description of natural languages. The SUBCATEGORIZATION PRINCIPLE ensures the correct valency of projections of lexical functors when (some of) their arguments are realized as syntactic daughters in phrasal structures: In a headed phrase, the list value of DAUGHTERS | HEAD-DAUGHTER | SYNSEM | LOCAL | CATEGORY | SUBCAT is the concatenation of the list value of SYNSEM | LOCAL | CATEGORY | SUBCAT with the list consisting of the SYNSEM values (in order) of the elements of the list value of DAUGHTERS | COMPLEMENT-DAUGHTERS [Pollard and Sag, 1994, p. 399].

(1) *The* SUBCATEGORIZATION PRINCIPLE *in AVM notation*

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS} \ \ \textit{headed-struc} \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} \textit{phrase} & \\ \text{SS LOC CAT SUBCAT} \ \boxed{1} & \\ \text{DTRS} & \begin{bmatrix} \textit{headed-struc} & \\ \text{HEAD-DTR} & \begin{bmatrix} \textit{sign} \\ \text{SS LOC CAT SUBCAT} \ \boxed{3} \end{bmatrix} \\ \text{COMP-DTRS} \ \boxed{2} & \end{bmatrix} \end{bmatrix}$$

$$\land \ \texttt{sign-ss-append}(\boxed{1},\boxed{2},\boxed{3})$$

The relation `sign-ss-append` is supposed to ensure the correct relationship between the SUBCAT list of the phrase (a list of *synsems*), the list of complement daughters (a list of *signs*), and the SUBCAT list of the head daughter (again a list of *synsems*) as verbosely described by Pollard and Sag. The relation `sign-ss-append` is clearly a close relative of the relation `append`, but the slight variation indicates the flexibility that grammarians expect to be given by the logical formalism for their use of relations. Note that the AVM notation in (1) appeals to the existential closure convention that we introduced on the handout of June 15th. We will observe appeals to this convention throughout many examples of principles without pointing them out any more.

The ID Principle is in many respects HPSG's counterpart to the phrase structure rules of classical phrase structure grammars and the phrase structural base of various versions of transformational grammar. The purpose of the ID Principle is to restrict the structural makeup of headed phrases in the same way as phrase structure rules dictate the possible nature of syntactic daughters. According to the ID Principle every headed phrase must satisfy exactly one of the ID schemata [Pollard and Sag, 1994, p. 399 and p. 402–403]. Pollard and Sag assume that there are 6 ID schemata for English. Schema 6, which we want to investigate more closely, is the Head-Filler Schema: The DAUGHTERS value is an object of sort *head-filler-struc* whose HEAD-DAUGHTER | SYNSEM | LOCAL | CATEGORY value satisfies the description [HEAD *verb*[VFORM *finite*], SUBCAT ⟨⟩], whose HEAD-DAUGHTER | SYNSEM | NONLOCAL | INHERITED | SLASH value contains an element token-identical to the FILLER-DAUGHTER | SYNSEM | LOCAL value, and whose HEAD-DAUGHTER | SYNSEM | NONLOCAL | TO-BIND | SLASH value contains only that element [Pollard and Sag, 1994, p. 403]. In (2), we see a schematic representation of the ID Principle, where the AVM formulae that express the 6 ID SCHEMATA are represented by 6 place holders in a disjunction in the consequent of the principle. The AVM formula for Schema6, the Head-Filler Schema, is then spelled out.

(2) *The* ID Principle

$$\left[\text{DTRS}\ \textit{headed-struc}\right] \rightarrow$$

$$(\text{Schema1} \vee \text{Schema2} \vee \text{Schema3} \vee \text{Schema4} \vee \text{Schema5} \vee \text{Schema6})$$

where Schema6 stands for

$$\left[\text{DTRS}\ \left[\begin{array}{l} \textit{head-filler-struc} \\ \text{FILLER-DTR SS LOC } \boxed{1} \\ \text{HEAD-DTR SS} \left[\begin{array}{l} \text{LOC CAT} \left[\begin{array}{l}\text{HEAD VFORM } \textit{finite} \\ \text{SUBCAT } \langle\rangle\end{array}\right] \\ \text{NONLOC} \left[\begin{array}{l}\text{INHERITED SLASH } \boxed{2} \\ \text{TO-BIND SLASH } \{\boxed{1}\}\end{array}\right]\end{array}\right]\end{array}\right]\right]$$
$$\wedge\ \texttt{member}(\boxed{1},\boxed{2})$$

The interesting feature of the Head-Filler Schema is the use of the `member` relation, expressing the membership of the filler's LOCAL value in the INHERITED SLASH set of the head daughter.

The most complex example of a principle that relies on relations in the theory of English of Pollard and Sag is their Binding Theory. It is summarized in their binding principles A–C:

(3) *The* Binding Theory *of [Pollard and Sag, 1994, p. 401] informally stated*

Principle A. A locally o-commanded anaphor must be locally o-bound.
Principle B. A personal pronoun must be locally o-free.
Principle C. A nonpronoun must be o-free.

Without going into any details, we observe that the BINDING THEORY is expressed in terms of relations (o-command, (local) o-binding, (local) o-freedom) that must or must not hold for different kinds of pronouns in syntactic structures. Bearing in mind that (local) o-freedom is described by Pollard and Sag as holding between two elements if and only if (local) o-binding does not hold, their binding principles can be rendered as AVM descriptions in the following way:

(4) *The* BINDING THEORY *in AVM notation*

Principle A:

$$\forall x\ \left(\exists y\ \mathtt{loc\text{-}o\text{-}command}(y, {}^{x}[\textsc{loc cont}\ \ ana]) \rightarrow \exists z\ \mathtt{loc\text{-}o\text{-}bind}(z, x)\right)$$

Principle B:

$$\forall x\ \left({}^{x}[\textsc{loc cont}\ \ ppro] \rightarrow \neg\exists y\ \mathtt{loc\text{-}o\text{-}bind}(y, x)\right)$$

Principle C:

$$\forall x\ \left({}^{x}[\textsc{loc cont}\ \ npro] \rightarrow \neg\exists y\ \mathtt{o\text{-}bind}(y, x)\right)$$

Independent of the linguistic significance of these principles, we can observe the full range of interaction between quantification, negation and relational expressions in them. Moreover, the relations of the BINDING THEORY are idiosyncratic, linguistically motivated relations that, unlike `member` and `append`, do not belong to the standard inventory of logical languages or of programming languages in computer science. Again we have to conclude that linguists require great freedom in using relational expressions with idiosyncratic meanings that they want to be able to define themselves in their grammars. Their relations are supposed to express linguistic generalizations about languages, and as such their formulation is subject to linguistic research and should not be limited by the formal tools used for that research. It does not seem to be sufficient to provide an inventory of standard relations with their corresponding meaning as part of the logical formalism itself, since it might not provide what a linguist needs for a particular descriptive purpose. The formalism has to provide the means to define the meaning of relation symbols as part of linguistic grammars to fit the linguistic needs of grammarians.

The language and model theory that we define in the following two sections corresponds almost completely to a feature structure based version of the RSRL formalism of [Richter, 2000]. For simplicity, it omits *chains*, a more flexible interpretation of variables which allows for expressing certain grammatical principles which employ relations with quasi-list arguments more directly than the restricted version of RSRL presented here.

1 Syntax

In order to work with relational expressions, we need to extend the initial signatures that we have been working with up to now. Just like attribute symbols, relation symbols have to be declared in signatures. In contrast to attributes, it is not sufficient to just declare the names of the relation symbols. We also need to say how many arguments each relation has. For example, the `member` relation usually has two arguments (the entity given as first argument is a member of the set given in the second argument),[1] whereas the standard `append` relation of

[1]Occasionally variants of this occur, in particular in the HPSG literature. The `member` relation is also used to indicate membership of elements on lists.

grammars has three arguments (the concatenation of the two lists given as the first and second argument is the list in the third argument). In our definition of *signatures* in DEFINITION 1, the set of relation symbols is symbolized as $\mathcal{R}$, and the arity function, $\mathcal{AR}$, assigns each relation symbol a positive natural number as the arity of the respective relation.

Definition 1 *Σ is a* ***signature*** *iff*

Σ *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,*

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

$\mathcal{A}$ *is a set,*

$\mathcal{F}$ *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* $\alpha \in \mathcal{A}$,

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* $\mathcal{F}(\langle \sigma_2, \alpha \rangle) \sqsubseteq \mathcal{F}(\langle \sigma_1, \alpha \rangle)$,

$\mathcal{R}$ *is a finite set, and*

$\mathcal{AR}$ *is a total function from* $\mathcal{R}$ *to* $\mathbb{N}^+$.

We need new notational conventions to declare relations and their arity. Figure 1 illustrates how we enrich our familiar signature of birds and pets by the standard relations `member` and `append`. The arity of the relations is indicated by notating it behind the relation symbols, separated from them by a slash.

Based on the definition of signatures, we define boxes and formulae relative to a given signature. The only difference to the previous definition is the addition of relational expressions to Σ formulae. The new lines are marked by an asterisk (*) on the margin.

Definition 2 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, $\mathbb{BOX}^\Sigma$ *and* $\mathcal{D}^\Sigma$ *are the smallest sets such that*

for each $X \in \mathcal{VAR}^\$$,

$X \in \mathbb{BOX}^\Sigma$,

for each $\sigma \in \mathcal{G}$,

$\sigma \in \mathbb{BOX}^\Sigma$,

for each $\alpha \in \mathcal{A}$, *for each* $\delta \in \mathbb{BOX}^\Sigma$,

$\alpha : \delta \ \in \mathbb{BOX}^\Sigma$,

```
type_hierarchy
bot
  list
     nelist head:animal tail:list
     elist
  animal legs:number color:color
     bird legs:two
       parrot
       woodpecker
       canary
     pet legs:four
       cat
       dog
  number
     one
     two
     three
     four
  color
     green
     red
     yellow
     brown
relations
append/3
member/2
.
```

Figure 1: A signature with relations in MoMo notation

for each $\delta \in \mathbb{BOX}^{\Sigma}$, $\sim \delta \ \in \mathbb{BOX}^{\Sigma}$,

for each $\delta_1 \in \mathbb{BOX}^{\Sigma}$, *for each* $\delta_2 \in \mathbb{BOX}^{\Sigma}$, $(\delta_1, \delta_2) \in \mathbb{BOX}^{\Sigma}$,

for each $\delta_1 \in \mathbb{BOX}^{\Sigma}$, *for each* $\delta_2 \in \mathbb{BOX}^{\Sigma}$, $(\delta_1; \delta_2) \in \mathbb{BOX}^{\Sigma}$,

for each $\delta_1 \in \mathbb{BOX}^{\Sigma}$, *for each* $\delta_2 \in \mathbb{BOX}^{\Sigma}$, $(\delta_1\texttt{*>}\delta_2) \in \mathbb{BOX}^{\Sigma}$,

for each $\delta_1 \in \mathbb{BOX}^{\Sigma}$, *for each* $\delta_2 \in \mathbb{BOX}^{\Sigma}$, $(\delta_1\texttt{<*>}\delta_2) \in \mathbb{BOX}^{\Sigma}$,

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

$$\delta \in \mathcal{D}^{\Sigma},$$

for each $X \in \mathcal{VAR}^{\$}$, *for each* $\delta \in \mathbb{BOX}^{\Sigma}$,

$$X : \delta \in \mathcal{D}^{\Sigma},$$

for each $X \in \mathcal{VAR}^{\$}$, for each $Y \in \mathcal{VAR}^{\$}$,

$X = Y \in \mathcal{D}^{\Sigma}$,

* for each $\rho \in \mathcal{R}$, for each $X_1 \in \mathcal{VAR}$, $\ldots$, for each $X_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,

$\rho\left(X_1, \ldots, X_{\mathcal{AR}(\rho)}\right) \in \mathcal{D}^{\Sigma}$,

for each $X \in \mathcal{VAR}$, for each $\delta \in \mathcal{D}^{\Sigma}$,

$\exists X(\delta) \in \mathcal{D}^{\Sigma}$,

for each $X \in \mathcal{VAR}$, for each $\delta \in \mathcal{D}^{\Sigma}$,

$\forall X(\delta) \in \mathcal{D}^{\Sigma}$,

for each $\delta \in \mathcal{D}^{\Sigma}$, $\sim \delta \in \mathcal{D}^{\Sigma}$,
for each $\delta_1 \in \mathcal{D}^{\Sigma}$, for each $\delta_2 \in \mathcal{D}^{\Sigma}$, $(\delta_1, \delta_2) \in \mathcal{D}^{\Sigma}$,
for each $\delta_1 \in \mathcal{D}^{\Sigma}$, for each $\delta_2 \in \mathcal{D}^{\Sigma}$, $(\delta_1; \delta_2) \in \mathcal{D}^{\Sigma}$,
for each $\delta_1 \in \mathcal{D}^{\Sigma}$, for each $\delta_2 \in \mathcal{D}^{\Sigma}$, $(\delta_1 \texttt{*>} \delta_2) \in \mathcal{D}^{\Sigma}$, and
for each $\delta_1 \in \mathcal{D}^{\Sigma}$, for each $\delta_2 \in \mathcal{D}^{\Sigma}$, $(\delta_1 \texttt{<*>} \delta_2) \in \mathcal{D}^{\Sigma}$.

I call each element of $\mathbb{BOX}^{\Sigma}$ a Σ *box*, and each element of $\mathcal{D}^{\Sigma}$ a Σ *formula*. The new Σ formulae of the form $\rho\left(X_1, \ldots, X_{\mathcal{AR}(\rho)}\right)$ consist of a relation symbol together with a number variables corresponding to the arity of the relation within round brackets. I call them relational Σ formulae.

Just as with complex grammars, our final notion of grammars will not use arbitrary formulae but only that subset of formulae that do not contain any unbound occurrences of variables. We formalize this idea using the function FV. Definition 3 differs from its preliminary version on the previous handout only in the additional base case for relational Σ formulae.

Definition 3 *For each signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, FV is the total function from $\mathbb{BOX}^{\Sigma} \cup \mathcal{D}^{\Sigma}$ to $Pow(\mathcal{VAR})$ such that*

$FV(\$) = \emptyset$,
for each $X \in \mathcal{VAR}$, $FV(X) = \{X\}$,
for each $\sigma \in \mathcal{G}$, $FV(\sigma) = \emptyset$,
for each $\alpha \in \mathcal{A}$, for each $\delta \in \mathbb{BOX}^{\Sigma}$, $FV(\alpha : \delta) = FV(\delta)$,
for each $X \in \mathcal{VAR}^{\$}$, for each $\delta \in \mathbb{BOX}^{\Sigma}$, $FV(X : \delta) = FV(X) \cup FV(\delta)$,
for each $X \in \mathcal{VAR}^{\$}$, for each $Y \in \mathcal{VAR}^{\$}$, $FV(X = Y) = FV(X) \cup FV(Y)$,
for each $\rho \in \mathcal{R}$, for each $X_1 \in \mathcal{VAR}$, $\ldots$, for each $X_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,
$FV\left(\rho\left(X_1, \ldots, X_{\mathcal{AR}(\rho)}\right)\right) = FV(X_1) \cup \ldots \cup FV\left(X_{\mathcal{AR}(\rho)}\right)$,
for each $X \in \mathcal{VAR}$, for each $\delta \in \mathcal{D}^{\Sigma}$, $FV(\exists X(\delta)) = FV(\delta) \backslash \{X\}$,
for each $X \in \mathcal{VAR}$, for each $\delta \in \mathcal{D}^{\Sigma}$, $FV(\forall X(\delta)) = FV(\delta) \backslash \{X\}$,
for each $\delta \in \mathbb{BOX}^{\Sigma} \cup \mathcal{D}^{\Sigma}$, $FV(\sim \delta) = FV(\delta)$,

for each $\delta_1 \in \mathbb{BOX}^\Sigma \cup \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathbb{BOX}^\Sigma \cup \mathcal{D}^\Sigma$, $FV((\delta_1, \delta_2)) = FV(\delta_1) \cup FV(\delta_2)$,
for each $\delta_1 \in \mathbb{BOX}^\Sigma \cup \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathbb{BOX}^\Sigma \cup \mathcal{D}^\Sigma$, $FV((\delta_1; \delta_2)) = FV(\delta_1) \cup FV(\delta_2)$,
for each $\delta_1 \in \mathbb{BOX}^\Sigma \cup \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathbb{BOX}^\Sigma \cup \mathcal{D}^\Sigma$, $FV((\delta_1 \texttt{*>} \delta_2)) = FV(\delta_1) \cup FV(\delta_2)$,
for each $\delta_1 \in \mathbb{BOX}^\Sigma \cup \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathbb{BOX}^\Sigma \cup \mathcal{D}^\Sigma$, $FV((\delta_1 \texttt{<*>} \delta_2)) = FV(\delta_1) \cup FV(\delta_2)$.

The function FV yields for each Σ box and for each Σ formula the set of variables that occur free in them. For grammars we are interested in Σ *descriptions*, the set of Σ formulae that do not contain any unbound variables:

Definition 4 *For each signature* Σ,

$$\mathcal{D}_0^\Sigma = \{\delta \in \mathcal{D}^\Sigma \mid FV(\delta) = \emptyset\}.$$

For each signature Σ, I call each member of $\mathcal{D}_0^\Sigma$ a Σ *description*. Our final notion of a grammar is based on signatures and on Σ descriptions:

Definition 5 Γ *is an* ***grammar*** *iff*

Γ *is a pair* $\langle \Sigma, \theta \rangle$,

Σ *is a signature, and*

$\theta \subseteq \mathcal{D}_0^\Sigma$.

A grammar is a signature, Σ, together with a set of Σ descriptions.

2 Meaning

The real work of a relational extension of a feature logic with an interpretation in a domain of feature structures happens in finding an appropriate extension of the notion of feature structures. Standard definitions of feature structures simply do not foresee a representation of arbitrary relations. But this is exactly what we need in order to be able to say what it means for a feature structure to satisfy a description that contains a relational formula.

DEFINITION 6 extends simple abstract feature structures (triples consisting of a basis set, a re-entrancy relation, and a label function) to quadruples. I call the extensions *relational abstract feature structures*. They comprise all components of simple abstract feature structures, defined in exactly the same way as before; plus, additionally, a relation extension, ξ, whose elements are tuples of relation symbols together with paths. The idea behind that representation is that the paths indicate the abstract nodes that stand in the relation. For each relation symbol ρ of the signature, each relational abstract feature structure contains tuples in ξ in which the (abstract) nodes that are in ρ are represented as the paths π that lead to them. This leads to the form $\langle \rho, \pi_1, \ldots, \pi_{\mathcal{AR}(\rho)} \rangle$ of these tuples, in which the symbol ρ indicates to which relation the following $\mathcal{AR}(\rho)$-tuple of paths belongs, and each $\mathcal{AR}(\rho)$-tuple of paths indicates which abstract nodes are a tuple of arguments in the relation ρ.

Definition 6 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, $\mathbb{A}$ *is a* ***relational abstract feature structure under*** Σ *iff*

$\mathbb{A}$ *is a quadruple* $\langle \beta, \varrho, \lambda, \xi \rangle$,

$\beta \subseteq \mathcal{A}^*$,

$\varepsilon \in \beta$,

for each $\pi \in \mathcal{A}^*$, *for each* $\alpha \in \mathcal{A}$,

if $\pi\alpha \in \beta$ *then* $\pi \in \beta$,

ϱ *is an equivalence relation over* β,

for each $\pi_1 \in \mathcal{A}^*$, *for each* $\pi_2 \in \mathcal{A}^*$, *for each* $\alpha \in \mathcal{A}$,

if $\pi_1\alpha \in \beta$ *and* $\langle \pi_1, \pi_2 \rangle \in \varrho$ *then* $\langle \pi_1\alpha, \pi_2\alpha \rangle \in \varrho$,

λ *is a total function from* β *to* $\mathcal{S}$, *and*

for each $\pi_1 \in \mathcal{A}^*$, *for each* $\pi_2 \in \mathcal{A}^*$,

if $\langle \pi_1, \pi_2 \rangle \in \varrho$ *then* $\lambda(\pi_1) = \lambda(\pi_2)$.

for each $\pi \in \mathcal{A}^*$, *for each* $\alpha \in \mathcal{A}$,

if $\pi\alpha \in \beta$ *then* $\mathcal{F}(\langle \lambda(\pi), \alpha \rangle)$ *is defined and* $\lambda(\pi\alpha) \sqsubseteq \mathcal{F}(\langle \lambda(\pi), \alpha \rangle)$,

for each $\pi \in \mathcal{A}^*$, *for each* $\alpha \in \mathcal{A}$,

if $\pi \in \beta$ *and* $\mathcal{F}(\langle \lambda(\pi), \alpha \rangle)$ *is defined then* $\pi\alpha \in \beta$,

$\xi \subseteq \mathcal{R} \times \beta^*$,

for each $\rho \in \mathcal{R}$, *for each* $\pi_1 \in \beta$,..., *for each* $\pi_n \in \beta$,

if $\langle \rho, \pi_1, \ldots, \pi_n \rangle \in \xi$ *then* $n = \mathcal{AR}(\rho)$, *and*

for each $\rho \in \mathcal{R}$, *for each* $\pi_1 \in \beta$, ..., *for each* $\pi_n \in \beta$, *for each* $\pi'_1 \in \beta$, ..., *for each* $\pi'_n \in \beta$,

if $\langle \rho, \pi_1, \ldots, \pi_n \rangle \in \xi$, *and*
for each $i \in \{1, \ldots, n\}$,
$\pi_i \in \beta$ *and* $\langle \pi_i, \pi'_i \rangle \in \varrho$,
then $\langle \rho, \pi'_1, \ldots, \pi'_n \rangle \in \xi$.

The last condition (in the last 6 lines of DEFINITION 6) on the tuples in ξ is saying that if some path π occurs in the nth argument position of a relation then all other paths π' that lead to the same abstract node must also occur in that position in all possible combinations of all other paths representing the other argument positions.

In the final version of the satisfaction functions we need to refer to Σ insertions and some version of the function *VarP* that was used to find the paths that lead to the occurrences of variables in Σ boxes. Fortunately, as we can find out by inspection, we can minimally modify the old definitions by changing the reference to initial signatures to signatures. Since the new components of signatures, $\mathcal{R}$ and $\mathcal{AR}$, are nowhere refered to in the definitions, they do not change in any significant way. What we have to modify, though, is the definition of π reducts of feature structures, since we need to say what the π reduct of the representations of relations in the relational extension of feature structures is. To get the π reduct of ξ, we take each tuple in it and remove the initial string π from each path in the paths representing the nodes that stand in the relation.

Definition 7 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, *for each* $\mathbb{A} = \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma}$, *for each* $\pi \in \mathcal{A}^*$,

$$\begin{aligned}
&\beta/\pi = \{\pi' \in \mathcal{A}^* \mid \pi\pi' \in \beta\},\\
&\varrho/\pi = \{\langle \pi_1, \pi_2 \rangle \in \mathcal{A}^* \times \mathcal{A}^* \mid \langle \pi\pi_1, \pi\pi_2 \rangle \in \varrho\},\\
&\lambda/\pi = \{\langle \pi', \sigma \rangle \in \mathcal{A}^* \times \mathcal{S} \mid \langle \pi\pi', \sigma \rangle \in \lambda\},\\
&\xi/\pi = \{\langle \rho, \pi_1, \ldots, \pi_{\mathcal{AR}(\rho)} \rangle \in \mathcal{R} \times (\beta/\pi)^*\} \mid \langle \rho, \pi\pi_1, \ldots, \pi\pi_{\mathcal{AR}(\rho)} \in \xi \rangle\}, \textit{ and}\\
&\mathbb{A}/\pi = \langle \beta/\pi, \varrho/\pi, \lambda/\pi, \xi/\pi \rangle.
\end{aligned}$$

As before, we can show that if a relational abstract feature structure under some signature Σ, $\mathbb{A}$, contains path π as an element of its basis set then the π reduct of $\mathbb{A}$ is also a relational abstract feature structure under Σ.

We can now immediately proceed to the final definition of the satisfaction functions. For each signature Σ and each Σ insertion $\iota \in \mathrm{I}_{\Sigma}$, the relational abstract feature structure satisfaction function under ι in Σ, Δ^{ι}_{Σ}, is defined inductively over all Σ formulae. As in the previous version, the definition of the semantics for Σ formulae follows the syntactic structure of our expressions and uses a second feature structure satisfaction function under ι in Σ for Σ boxes, Ξ^{ι}_{Σ}, which gives us the relational feature structure denotation of Σ boxes under ι. Since the structure of Σ boxes has not changed at all, Ξ^{ι}_{Σ} does change either. The only difference to the previous satisfaction functions is the additional clause for relational Σ formulae.

Definition 8 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, *for each* Σ *insertion* $\iota \in \mathrm{I}_{\Sigma}$, Ξ^{ι}_{Σ} *is the total function from* $\mathbb{BOX}^{\Sigma}$ *to* $Pow(\mathbb{AFS}^{\Sigma})$ *and* Δ^{ι}_{Σ} *is the total function from* $\mathcal{D}^{\Sigma}$ *to* $Pow(\mathbb{AFS}^{\Sigma})$ *such that*

for each $X \in \mathcal{VAR}^{\$}$,

$$\Xi^{\iota}_{\Sigma}(X) = \mathbb{AFS}^{\Sigma},$$

for each $\sigma \in \mathcal{G}$,

$$\Xi^{\iota}_{\Sigma}(\sigma) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \;\middle|\; \begin{array}{l} \textit{for some } \sigma' \in \mathcal{S}, \\ \quad \langle \varepsilon, \sigma' \rangle \in \lambda, \textit{ and} \\ \quad \sigma' \sqsubseteq \sigma \end{array} \right\},$$

for each $\alpha \in \mathcal{A}$, *for each* $\delta \in \mathbb{BOX}^{\Sigma}$,

$$\Xi^{\iota}_{\Sigma}(\alpha : \delta) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \;\middle|\; \begin{array}{l} \alpha \in \beta, \textit{ and} \\ \langle \beta, \varrho, \lambda, \xi \rangle/\alpha \in \Xi^{\iota}_{\Sigma}(\delta) \end{array} \right\},$$

for each $\delta \in \mathbb{BOX}^{\Sigma}$, $\Xi^{\iota}_{\Sigma}(\sim \delta) = \mathbb{AFS}^{\Sigma} \backslash \Xi^{\iota}_{\Sigma}(\delta)$,

for each $\delta_1 \in \mathbb{BOX}^{\Sigma}$, *for each* $\delta_2 \in \mathbb{BOX}^{\Sigma}$, $\Xi^{\iota}_{\Sigma}(\delta_1, \delta_2) = \Xi^{\iota}_{\Sigma}(\delta_1) \cap \Xi^{\iota}_{\Sigma}(\delta_2)$,

for each $\delta_1 \in \mathbb{BOX}^{\Sigma}$, *for each* $\delta_2 \in \mathbb{BOX}^{\Sigma}$, $\Xi^{\iota}_{\Sigma}(\delta_1 ; \delta_2) = \Xi^{\iota}_{\Sigma}(\delta_1) \cup \Xi^{\iota}_{\Sigma}(\delta_2)$,

for each $\delta_1 \in \mathbb{BOX}^{\Sigma}$, *for each* $\delta_2 \in \mathbb{BOX}^{\Sigma}$, $\Xi^{\iota}_{\Sigma}(\delta_1 \texttt{*>} \delta_2) = \left(\mathbb{AFS}^{\Sigma} \backslash \Xi^{\iota}_{\Sigma}(\delta_1)\right) \cup \Xi^{\iota}_{\Sigma}(\delta_2)$,

for each $\delta_1 \in \mathbb{BOX}^{\Sigma}$, *for each* $\delta_2 \in \mathbb{BOX}^{\Sigma}$,
$\Xi^{\iota}_{\Sigma}(\delta_1 \texttt{<*>} \delta_2) = \left(\left(\mathbb{AFS}^{\Sigma} \backslash \Xi^{\iota}_{\Sigma}(\delta_1)\right) \cap \left(\mathbb{AFS}^{\Sigma} \backslash \Xi^{\iota}_{\Sigma}(\delta_2)\right)\right) \cup \left(\Xi^{\iota}_{\Sigma}(\delta_1) \cap \Xi^{\iota}_{\Sigma}(\delta_2)\right)$, *and*

for each $X \in \mathcal{VAR}^{\$}$,

$$\Delta^{\iota}_{\Sigma}(X) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \iota(X) \in \beta \right\},$$

for each $\sigma \in \mathcal{G}$,

$$\Delta^{\iota}_{\Sigma}(\sigma) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \begin{array}{l} \textit{for some } \sigma' \in \mathcal{S}, \\ \quad \langle \varepsilon, \sigma' \rangle \in \lambda, \textit{ and} \\ \quad \sigma' \sqsubseteq \sigma \end{array} \right\},$$

for each $\alpha \in \mathcal{A}$, *for each* $\delta \in \mathbb{BOX}^{\Sigma}$,

$$\Delta^{\iota}_{\Sigma}(\alpha : \delta) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \begin{array}{l} \alpha \in \beta, \\ \langle \beta, \varrho, \lambda, \xi \rangle / \alpha \in \Xi^{\iota}_{\Sigma}(\delta), \textit{ and} \\ \textit{for each } \langle \pi, X \rangle \in \textit{VarP}(\alpha, \delta), \\ \quad \langle \pi, \iota(X) \rangle \in \varrho \end{array} \right\},$$

for each $X \in \mathcal{VAR}^{\$}$, *for each* $\delta \in \mathbb{BOX}^{\Sigma}$,

$$\Delta^{\iota}_{\Sigma}(X : \delta) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \begin{array}{l} \iota(X) \in \beta, \textit{ and} \\ \langle \beta, \varrho, \lambda, \xi \rangle / \iota(X) \in \Xi^{\iota}_{\Sigma}(\delta), \textit{ and} \\ \textit{for each } \langle \pi, Y \rangle \in \textit{VarP}(\iota(X), \delta), \\ \quad \langle \pi, \iota(Y) \rangle \in \varrho \end{array} \right\},$$

for each $X \in \mathcal{VAR}^{\$}$, *for each* $Y \in \mathcal{VAR}^{\$}$,

$$\Delta^{\iota}_{\Sigma}(X = Y) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \langle \iota(X), \iota(Y) \rangle \in \varrho \right\},$$

for each $\rho \in \mathcal{R}$, *for each* $X_1 \in \mathcal{VAR}$, $\ldots$, *for each* $X_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,

$$\Delta^{\iota}_{\Sigma}(\rho(X_1, \ldots, X_{\mathcal{AR}(\rho)})) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \langle \rho, \iota(X_1), \ldots, \iota(X_{\mathcal{AR}(\rho)}) \rangle \in \xi \right\},$$

for each $X \in \mathcal{VAR}$, *for each* $\delta \in \mathcal{D}^{\Sigma}$,

$$\Delta^{\iota}_{\Sigma}(\exists X(\delta)) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \begin{array}{l} \textit{for some } \pi \in \beta, \\ \quad \langle \beta, \varrho, \lambda, \xi \rangle \in \Delta^{\iota[\frac{\pi}{X}]}_{\Sigma}(\delta) \end{array} \right\},$$

for each $X \in \mathcal{VAR}$, *for each* $\delta \in \mathcal{D}^{\Sigma}$,

$$\Delta^{\iota}_{\Sigma}(\forall X(\delta)) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \begin{array}{l} \textit{for each } \pi \in \beta, \\ \quad \langle \beta, \varrho, \lambda, \xi \rangle \in \Delta^{\iota[\frac{\pi}{X}]}_{\Sigma}(\delta) \end{array} \right\},$$

for each $\delta \in \mathcal{D}^{\Sigma}$, $\Delta^{\iota}_{\Sigma}(\sim \delta) = \mathbb{AFS}^{\Sigma} \backslash \Delta^{\iota}_{\Sigma}(\delta)$,

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$, $\Delta^{\iota}_{\Sigma}(\delta_1, \delta_2) = \Delta^{\iota}_{\Sigma}(\delta_1) \cap \Delta^{\iota}_{\Sigma}(\delta_2)$,

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$, $\Delta^{\iota}_{\Sigma}(\delta_1; \delta_2) = \Delta^{\iota}_{\Sigma}(\delta_1) \cup \Delta^{\iota}_{\Sigma}(\delta_2)$,

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$, $\Delta^{\iota}_{\Sigma}(\delta_1 \texttt{*>} \delta_2) = \left(\mathbb{AFS}^{\Sigma} \backslash \Delta^{\iota}_{\Sigma}(\delta_1)\right) \cup \Delta^{\iota}_{\Sigma}(\delta_2)$, *and*

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,
$$\Delta^{\iota}_{\Sigma}(\delta_1 \texttt{<*>} \delta_2) = \left(\left(\mathbb{AFS}^{\Sigma} \backslash \Delta^{\iota}_{\Sigma}(\delta_1)\right) \cap \left(\mathbb{AFS}^{\Sigma} \backslash \Delta^{\iota}_{\Sigma}(\delta_2)\right)\right) \cup \left(\Delta^{\iota}_{\Sigma}(\delta_1) \cap \Delta^{\iota}_{\Sigma}(\delta_2)\right).$$

I call Δ^{ι}_{Σ} the *relational abstract feature structure satisfaction function under* ι *in* Σ, and say $\mathbb{A}$ satisfies δ under ι in Σ if and only if $\mathbb{A} \in \Delta^{\iota}_{\Sigma}(\delta)$. Intuitively speaking, a relational feature structure satisfies a relational expression if and only if the paths inserted for the variables in the relational expression lead to abstract nodes that stand in the relation according to the representation of the relation given by the elements of ξ.

Definition 9 *For each signature* Σ,

M_{Σ} *is the total function from* $Pow(\mathcal{D}^{\Sigma}_{0})$ *to* $Pow(\mathbb{AFS}^{\Sigma})$ *such that for each* $\theta \subseteq \mathcal{D}^{\Sigma}_{0}$,

$$M_{\Sigma}(\theta) = \left\{ \langle \beta, \varrho, \lambda, \xi \rangle \in \mathbb{AFS}^{\Sigma} \,\middle|\, \begin{array}{l} \textit{for each } \pi \in \beta, \\ \textit{for each } \iota \in \mathrm{I}_{\Sigma}, \textit{ and} \\ \textit{for each } \delta \in \theta, \\ \quad \langle \beta, \varrho, \lambda, \xi \rangle / \pi \in \Delta^{\iota}_{\Sigma}(\delta) \end{array} \right\}.$$

For each signature Σ, I call M_{Σ} the *relational abstract feature structure admission function in* Σ. A simple abstract feature structure under Σ, $\mathbb{A}$, is in the set of relational abstract feature structures under Σ admitted by a Σ theory θ exactly if $\mathbb{A}$ and every possible π reduct of $\mathbb{A}$ satisfy every description in θ. Except for the extended notion of abstract feature structures, this is the definition that we already had before with simpler description languages.

Given a grammar, $\langle \Sigma, \theta \rangle$, we regard the set of relational abstract feature structures admitted by θ, $M_{\Sigma}(\theta)$, as the intended meaning of $\langle \Sigma, \theta \rangle$. Pollard and Sag say that the members of $M_{\Sigma}(\theta)$ are mathematical representations of the object types of the language that the grammar is about. Our syntax and semantics thus gives us a complete formalization of HPSG as it was presented informally in [Pollard and Sag, 1994].

References

[Pollard and Sag, 1994] Pollard, Carl and Sag, Ivan A. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

[Richter, 2000] Richter, Frank 2000. *A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar*. Phil. dissertation, Eberhard-Karls-Universität Tübingen.