Seminar für Sprachwissenschaft
Abteilung Computerlinguistk
Universität Tübingen

SS 2002
June 19th, 2002

Grammar Formalisms and Parsing

Prof. Dr. E. Hinrichs
Frank Richter
Beata Trawiński

Having discussed a basic version of the syntax and semantics of HPSG grammars, we now turn to an extension of the description language. This extension will introduce variables and a special form of universal and existential quantification. The extension of the basic formal language will make it possible to express the identity (or inequality) of substructures in principles like the Head Feature Principle, the Spec Principle and the Control Theory of [Pollard and Sag, 1994], which were beyond the expressive power of our initial formalism. To see what is at stake, we repeat these principles as they are stated in the book, followed by their formulation in an AVM notation.

The Head Feature Principle says that, in a headed phrase, the value of SYNSEM | LOCAL | CATEGORY | HEAD and DAUGHTERS | HEAD-DAUGHTER | SYNSEM | LOCAL | CATEGORY | HEAD are token-identical [Pollard and Sag, 1994, p. 399]:

(1) *The* Head Feature Principle *in AVM notation*

$$\begin{bmatrix} \textit{phrase} \\ \text{DTRS}\ \ \textit{headed-struc} \end{bmatrix} \rightarrow \begin{bmatrix} \text{SYNSEM LOC CAT HEAD} & \boxed{1} \\ \text{DTRS HEAD-DTR SYNSEM LOC CAT HEAD} & \boxed{1} \end{bmatrix}$$

The Spec Principle says that, in a headed phrase whose nonhead daughter (either the MARKER-DAUGHTER or COMPLEMENT-DAUGHTERS | FIRST) has a SYNSEM | LOCAL | CATEGORY | HEAD value of sort *functional*, the SPEC value of that value must be token-identical with the phrase's DAUGHTERS | HEAD-DAUGHTER | SYNSEM value [Pollard and Sag, 1994, p. 400]:

(2) *The* Spec Principle *in AVM notation*

$$\forall\boxed{1}\ \forall\boxed{2}\left(\left(\left[\text{DTRS}\ \left[\left[\text{MARKER-DTR}\ \boxed{1}\right] \vee \left[\text{COMP-DTRS}\ \langle \boxed{1} | \textit{list} \rangle\right]\right]\right] \wedge \boxed{1}\left[\text{SS LOC CAT HEAD}\ \begin{bmatrix} \textit{functional} \\ \text{SPEC}\ \boxed{2} \end{bmatrix}\right]\right) \rightarrow \left[\text{DTRS HEAD-DTR SS}\ \boxed{2}\right]\right)$$

The Control Theory says that, if the SOA-ARG value of a *control-qfpsoa* is token-identical with the CONTENT value of a *local* entity whose CATEGORY | SUBCAT value is a list of length one, then the member of that list is (1) reflexive, and (2) coindexed with the INFLUENCED (respectively, COMMITTOR, EXPERIENCER) value of the *control-qfpsoa* if the latter is of sort *influence* (respectively, *commitment*, *orientation*) [Pollard and Sag, 1994, p. 401]:

(3) *The* CONTROL THEORY *in AVM notation*

$$\forall\boxed{1}\ \forall\boxed{2}\ \forall\boxed{3}\left(\begin{array}{l}\exists\boxed{4}\left(\boxed{1}\begin{bmatrix}\textit{influence}\\ \text{SOA-ARG}\ \boxed{4}\end{bmatrix}\wedge\boxed{2}\begin{bmatrix}\textit{local}\\ \text{CAT SUBCAT}\ \langle\boxed{3}\rangle\\ \text{CONT}\ \boxed{4}\end{bmatrix}\right)\rightarrow\\ \exists\boxed{5}\left(\boxed{3}\begin{bmatrix}\text{LOC CONT}\ \begin{bmatrix}\textit{reflexive}\\ \text{INDEX}\ \boxed{5}\end{bmatrix}\end{bmatrix}\wedge\boxed{1}\begin{bmatrix}\text{INFLUENCED}\ \boxed{5}\end{bmatrix}\right)\end{array}\right)\wedge$$

$$\forall\boxed{1}\ \forall\boxed{2}\ \forall\boxed{3}\left(\begin{array}{l}\exists\boxed{4}\left(\boxed{1}\begin{bmatrix}\textit{commitment}\\ \text{SOA-ARG}\ \boxed{4}\end{bmatrix}\wedge\boxed{2}\begin{bmatrix}\textit{local}\\ \text{CAT SUBCAT}\ \langle\boxed{3}\rangle\\ \text{CONT}\ \boxed{4}\end{bmatrix}\right)\rightarrow\\ \exists\boxed{5}\left(\boxed{3}\begin{bmatrix}\text{LOC CONT}\ \begin{bmatrix}\textit{reflexive}\\ \text{INDEX}\ \boxed{5}\end{bmatrix}\end{bmatrix}\wedge\boxed{1}\begin{bmatrix}\text{COMMITTOR}\ \boxed{5}\end{bmatrix}\right)\end{array}\right)\wedge$$

$$\forall\boxed{1}\ \forall\boxed{2}\ \forall\boxed{3}\left(\begin{array}{l}\exists\boxed{4}\left(\boxed{1}\begin{bmatrix}\textit{orientation}\\ \text{SOA-ARG}\ \boxed{4}\end{bmatrix}\wedge\boxed{2}\begin{bmatrix}\textit{local}\\ \text{CAT SUBCAT}\ \langle\boxed{3}\rangle\\ \text{CONT}\ \boxed{4}\end{bmatrix}\right)\rightarrow\\ \exists\boxed{5}\left(\boxed{3}\begin{bmatrix}\text{LOC CONT}\ \begin{bmatrix}\textit{reflexive}\\ \text{INDEX}\ \boxed{5}\end{bmatrix}\end{bmatrix}\wedge\boxed{1}\begin{bmatrix}\text{ORIENTATION}\ \boxed{5}\end{bmatrix}\right)\end{array}\right)$$

The syntactic shape of the AVM notation of the principles in (1)–(3) reveals their extended syntax, which we have to integrate into our formal language.

1 Syntax

Our syntactic extension of the initial syntax of descriptions will still be based on initial signatures. The only additional symbols that we will use come from a countably infinite set of variables, $\mathcal{VAR}$, augmented by the reserved symbol \$. For simplicity, I write $\mathcal{VAR}^{\$}$ when I refer to the set $\mathcal{VAR} \cup \{\$\}$. I use capital latin letters like X, Y, and Z for the members of $\mathcal{VAR}$. \$ is a reserved symbol, which will be used as a special kind of variable.

Based on the definition of initial signatures, we define initial formulae relative to a given initial signature:

Definition 1 *For each initial signature $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F} \rangle$, $\mathbb{BOX}^{\Sigma}$ 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 $\delta \in \mathbb{BOX}^{\Sigma}$,

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

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 $X \in \mathcal{VAR}$, *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 $\mathcal{D}^{\Sigma}$ an *initial* Σ *formula*. Initial Σ formulae of the form $X = Y$, where X and Y are variables, are called equations. An expression of the form $X : \delta$, with X a variable and δ a Σ box, is called a tagged Σ box. If a variable occurs to the right of the colon, ':', in a Σ box or Σ formula of the form $\alpha : \delta$ or $X : \delta$, I say that it occurs in a Σ box. The only kind of syntactic expression missing in initial Σ formula compared to what we will finally need is relational expressions.

Initial Σ formulae are clearly more complex than the initial Σ descriptions of our first syntax. This is due to the fact that they are built from other possibly complex expressions, Σ boxes. Roughly speaking, Σ boxes correspond to the familiar AVM matrices, the expressions between angled brackets in the principles (1)–(3). More complex initial Σ formulae can be built out of them by prefixing them with variables (tags), combining expressions with the logical connectives, by quantifying over complex expressions, and by forming equations between variables. The special nature of the reserved symbol, \$, becomes clear by its syntactic distribution: In contrast to normal variables, it may not be quantified over.

Our new notion of grammars will not use arbitrary initial formulae but only a restricted subset of them. The relevant subset is the set of those initial formulae that do not contain any free occurrences of variables. A variable occurs free in an initial formula if and only if it is not in the scope of any quantifier binding the variable in the initial formula. We formalize this idea using the function FV:

Definition 2 *For each initial signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F} \rangle$, 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 $X \in \mathcal{VAR}$, *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 initial Σ formula the set of variables that occur free in them. For grammars we are interested in Σ *descriptions*, the set of initial Σ formulae that do not contain any unbound variables:

Definition 3 *For each initial signature* Σ,

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

For each initial signature Σ, I call each member of $\mathcal{D}_0^\Sigma$ a Σ *description*. Our new notion of a grammar is based on initial signatures and on Σ descriptions. In DEFINITION 4, we use the insight that the principles of an HPSG grammar are a set of descriptions. The Greek letter θ now refers to a set of Σ descriptions:

Definition 4 Γ *is an* ***complex grammar*** *iff*

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

Σ *is an initial signature, and*

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

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

2 Meaning

We keep our simple abstract feature structures to interpret grammars written with our new extended language. Before we can define a new satisfaction function between Σ descriptions and sets of simple abstract feature structures under Σ, however, we have to say how we want to interpret variables. Standard logical languages employ variable assignment functions that assign entities in the structures that serve as interpretations of the logical language to variables. In our case, the entities that serve as interpreting structures are abstract feature structures. Looking at the way HPSG uses tags, an adequate choice for the interpretation of the variables that formalize them are the nodes of feature structures. Since we use *abstract* feature structures, whose nodes are represented by an equivalence relation over sequences of attributes, it makes sense to assign sequences of attributes to variables. Naturally, the functions that do this are defined relative to initial signatures:

Definition 5 *For each initial signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F} \rangle$,

I_Σ *is the set of total functions from* $\mathcal{VAR}^{\$}$ *to* $\mathcal{A}^*$ *such that for each* $\iota \in \mathrm{I}_\Sigma$, $\iota(\$) = \varepsilon$.

For each initial signature Σ, I call each member of I_Σ a Σ *insertion*. The empty path, ε, is always inserted for the special symbol, \$.

Before we can define a satisfaction function for Σ boxes and Σ formulae we need to define one last auxiliary function. The purpose of the function *VarP* is to tell us for each Σ formula of the form $\alpha : \delta$ or $X : \delta$ which variables occur in the Σ box δ and which paths lead to them in the description. Informally speaking, our satisfaction function will then require that for a simple abstract feature structure to satisfy an expression each path lead to the same abstract node to which the corresponding variable refers.

Definition 6 *For each initial signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F} \rangle$, *VarP is the total function from the Cartesian product of* $\mathcal{A}^*$ *and* $\mathbb{BOX}^\Sigma$ *to* $Pow(\mathcal{A}^* \times \mathcal{VAR}^{\$})$ *such that*

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

$$VarP(\pi, X) = \{\langle \pi, X \rangle\},$$

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

$$VarP(\pi, \sigma) = \emptyset,$$

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

$$VarP(\pi, \alpha : \delta) = VarP(\pi\alpha, \delta),$$

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

$$VarP(\pi, \sim \delta) = VarP(\pi, \delta),$$

for each $\pi \in \mathcal{A}^*$, *for each* $\delta_1 \in \mathbb{BOX}^\Sigma$, *for each* $\delta_2 \in \mathbb{BOX}^\Sigma$,

$$VarP(\pi, (\delta_1, \delta_2)) = VarP(\pi, \delta_1) \cup VarP(\pi, \delta_2),$$

for each $\pi \in \mathcal{A}^*$, *for each* $\delta_1 \in \mathbb{BOX}^\Sigma$, *for each* $\delta_2 \in \mathbb{BOX}^\Sigma$,

$$VarP(\pi, (\delta_1; \delta_2)) = VarP(\pi, \delta_1) \cup VarP(\pi, \delta_2),$$

for each $\pi \in \mathcal{A}^*$, *for each* $\delta_1 \in \mathbb{BOX}^\Sigma$, *for each* $\delta_2 \in \mathbb{BOX}^\Sigma$,

$$VarP(\pi, (\delta_1 \texttt{*>} \delta_2)) = VarP(\pi, \delta_1) \cup VarP(\pi, \delta_2), \textit{ and}$$

for each $\pi \in \mathcal{A}^*$, *for each* $\delta_1 \in \mathbb{BOX}^\Sigma$, *for each* $\delta_2 \in \mathbb{BOX}^\Sigma$,

$$VarP(\pi, (\delta_1 \texttt{<*>} \delta_2)) = VarP(\pi, \delta_1) \cup VarP(\pi, \delta_2).$$

For each initial signature Σ and each Σ insertion $\iota \in \mathrm{I}_\Sigma$, the complex abstract feature structure satisfaction function under ι in Σ, Δ^ι_Σ, is defined inductively over all initial Σ formulae. The definition of the semantics for Σ formulae follows the syntactic structure of our complex expressions and uses a second feature structure satisfaction function under ι in Σ for Σ boxes, Ξ^ι_Σ, which gives us the feature structure denotation of Σ boxes under ι.

Definition 7 *For each initial signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F} \rangle$, *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 $\sigma \in \mathcal{G}$,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

I call Δ^{ι}_{Σ} the *simple abstract feature structure satisfaction function under* ι *in* Σ, and say $\mathbb{A}$ satisfies δ under ι in Σ if and only if $\mathbb{A} \in \Delta^{\iota}_{\Sigma}(\delta)$. Note that quantification is always quantification over the substructure of the (abstract) node to which the special symbol, \$, refers. It follows that existential and universal quantification in HPSG does not behave like existential and universal quantification in first order logic, where quantification ranges over the entire universe of entities. In terms of the feature structure models of HPSG grammars that we are considering here, quantification is always quantification within feature structures, never across different, structurally unrelated feature structures.

Definition 8 *For each initial signature* Σ*,*

M_Σ is the total function from $Pow(\mathcal{D}_0^\Sigma)$ to $Pow(\mathbb{AFS}^\Sigma)$ such that for each $\theta \subseteq \mathcal{D}_0^\Sigma$,

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

For each initial signature Σ, I call M_Σ the *simple abstract feature structure admission function in* Σ. A simple abstract feature structure under Σ, $\mathbb{A}$, is in the set of simple 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. This is precisely the definition that we already had before with our simpler description language.

Just as before, given a complex grammar, $\langle \Sigma, \theta \rangle$, we can regard the set of 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.

3 Notational conventions and examples

Complex grammars only allow descriptions as principles. This means that we have to quantify explicitly over all variables (tags) in all principles of grammar. In the linguistically oriented HPSG literature, this is typically not done. The difference between common practice and formal explicitness becomes clear when we look at the HEAD FEATURE PRINCIPLE in our current notation:

(4) *The* HEAD FEATURE PRINCIPLE *in our syntax*

$\exists X$ ((*phrase*, DTRS:*headed-struc*)*>

(SYNSEM:LOC:CAT:HEAD:X, DTRS:HEAD-DTR:SYNSEM:LOC:CAT:HEAD:X))

In order to be able to be a bit lazy with simple cases like the HEAD FEATURE PRINCIPLE, we introduce a notational convention: We will assume that whenever a variable in an initial Σ formula that is meant to be a principle of grammar is not explicitly bound by a quantifier, it is bound by an implicitly intended existential closure over the entire initial Σ formula. Note that this leads to wide scope existential quantification over the variables in question.

Convention 1 *If in a principle of grammar, a variable is not explicitly bound by a quantifier, we assume that it is captured by an implicit existential closure with wide scope over the entire initial* Σ *formula.*

With CONVENTION 1, we can simplify the notation of the HEAD FEATURE PRINCIPLE:

(5) *The* HEAD FEATURE PRINCIPLE *again*

((*phrase*, DTRS:*headed-struc*)*>

(SYNSEM:LOC:CAT:HEAD:X, DTRS:HEAD-DTR:SYNSEM:LOC:CAT:HEAD:X))

As a principle of grammar, (5) is a notational abbreviation of its fully explicit rendering in (4).

In complex principles of grammar it is necessary to be careful when we use this notational convention. In particular for beginners it is a good idea to write down all quantifiers explicitly if there is an interaction between several quantifiers in a principle.

4 Exercises

Exercise 1 *Translate the* SPEC PRINCIPLE, *(2), and the* CONTROL THEORY, *(3), into our syntax.*

Exercise 2 *Let Σ be our familiar initial signature of lists, birds, and pets. Write a complex grammar using that signature such that for feature structure models of your grammar the following conditions hold:*

1. *Each list contains at least one parrot.*
2. *There are at least two distinct canaries on each list.*
3. *Every list is finite.*

What is the difference between the simple abstract feature structures that satisfy the descriptions used to formalize (1) and (2) and the ones that model them?

Exercise 3 *Let Σ again be our familiar initial signature of lists, birds, and pets. Take a simple abstract feature structure under Σ representing a singleton list with a green parrot on it.*

Consider the following Σ descriptions and decide for each one whether our simple abstract feature structure under Σ satisfies it or is admitted by it. In the descriptions, `V` *stands for the universal quantifier, and* `^` *for the existential quantifier (MoMo notation for quantifiers).*

1. *(a)* `^X(X,color:green).`
   (b) `^X(color:green).`
2. *(a)* `Vx((X:parrot)*>(X:color:green)).`
   (b) `^X((X:parrot)*>(X:color:green)).`
3. *(a)* `VX((X:parrot)*>(color:green))`
   (b) `VX((X:parrot)*>(^Y(Y:color:yellow))).`

References

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