A Comparison of HPSG and LFG

Valia Kordoni, Frank Richter

ESSLLI 2000,
Birmingham,
8–18 August 2000

Formalization of a few principles of Pollard and Sag 1994

Throughout this handout, the syntax used in the formalization of principles of grammar is the AVM notation for RSRL defined in Section 3.2 of Richter 2000.

(1) *The* SUBJECT CONDITION *(Pollard and Sag, 1994, p. 400)*

If the initial element of the SUBCAT value of a word is slashed, then so is some other element of that list.

(2) *The* SUBJECT CONDITION *formalized*

$$\forall \boxed{1} \left(\begin{bmatrix} \textit{word} \\ \text{SS LOC CAT SUBCAT } \left\langle [\text{NONLOC INHERITED SLASH } \textit{neset}] \,|\, \boxed{1} \right\rangle \end{bmatrix} \rightarrow \exists x \ \texttt{member}\left({}^{x}[\text{NONLOC INHERITED SLASH } \textit{neset}], \boxed{1} \right) \right)$$

(3) *The* CONTROL THEORY *(Pollard and Sag, 1994, p. 401)*

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*).

(4) *The* CONTROL THEORY *formalized*

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

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

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

(5) *The* BINDING THEORY *(Pollard and Sag, 1994, p. 401)*

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.

The BINDING THEORY employs a number of auxiliary relations that we must define first. For constraining the meaning of relation symbols in the grammar, we employ a convenient clause notation.

> "One *synsem* object is *more oblique* than another provided it appears to the right of the other on the SUBCAT list of some word." (Pollard and Sag, 1994, p. 401)

(6) $\texttt{more-oblique}(x, y) \stackrel{\forall}{\Longleftarrow}$
$$ {}^{w}\begin{bmatrix} \textit{word} \\ \text{SS LOC CAT SUBCAT } \boxed{1} \end{bmatrix} \wedge \texttt{to-the-right}(x, y, \boxed{1}) $$

(7) $\texttt{to-the-right}(x, y, z) \stackrel{\forall}{\Longleftarrow}$
$$ \exists \boxed{1}\ (z \langle y | \boxed{1} \rangle \wedge \texttt{member}(x, \boxed{1})) $$
$$ \vee \exists \boxed{1}\ (z \langle \textit{object} | \boxed{1} \rangle \wedge \texttt{to-the-right}(x, y, \boxed{1})) $$

> "One referential *synsem* object *locally o-commands* another provided they have distinct LOCAL values and either (1) the second is more oblique than the first, or (2) the second is a member of the SUBCAT list of a *synsem* object that is more oblique than the first."

(8) $\texttt{loc-o-command}(x, y) \stackrel{\forall}{\Longleftarrow}$
$$ \begin{pmatrix} {}^{x}\Big[\text{LOC } \boxed{1} \big[\text{CONT INDEX } \textit{ref}\big]\Big] \wedge \\ {}^{y}\Big[\text{LOC } \boxed{2} \big[\text{CONT INDEX } \textit{ref}\big]\Big] \wedge \\ \neg \boxed{1} = \boxed{2} \end{pmatrix} \wedge $$
$$ \begin{pmatrix} \texttt{more-oblique}(y, x) \vee \\ \begin{pmatrix} \texttt{more-oblique}\left({}^{s}\begin{bmatrix} \textit{synsem} \\ \text{LOC CAT SUBCAT } \boxed{3} \end{bmatrix}, x \right) \wedge \\ \texttt{member}(y, \boxed{3}) \end{pmatrix} \end{pmatrix} $$

> "One referential *synsem* object *o-commands* another provided they have distinct LOCAL values and either (1) the second is more oblique than the first, (2) the second is a member of the SUBCAT list of a *synsem* object that is o-commanded by the first, or (3) the second has the same LOCAL | CATEGORY | HEAD value as a *synsem* object that is o-commanded by the first." (Pollard and Sag, 1994, p. 401)

(9) $\texttt{o-command}(x, y) \stackrel{\forall}{\Longleftarrow}$
$$ \begin{pmatrix} {}^{x}\Big[\text{LOC } \boxed{1} \big[\text{CONT INDEX } \textit{ref}\big]\Big] \wedge \\ {}^{y}\left[\text{LOC } \boxed{2} \begin{bmatrix} \text{CAT HEAD } \boxed{4} \\ \text{CONT INDEX } \textit{ref} \end{bmatrix}\right] \wedge \\ \neg \boxed{1} = \boxed{2} \end{pmatrix} \wedge $$
$$ \begin{pmatrix} \texttt{more-oblique}(y, x) \vee \\ \begin{pmatrix} \texttt{o-command}\left(x, {}^{s_1}\begin{bmatrix} \textit{synsem} \\ \text{LOC CAT SUBCAT } \boxed{3} \end{bmatrix}\right) \wedge \\ \texttt{member}(y, \boxed{3}) \end{pmatrix} \vee \\ \texttt{o-command}\left(x, {}^{s_2}\begin{bmatrix} \textit{synsem} \\ \text{LOC } \big[\text{CAT HEAD } \boxed{4}\big] \end{bmatrix}\right) \end{pmatrix} $$

> "One referential *synsem* object *(locally) o-binds* another provided it (locally) o-commands and is coindexed with the other. A referential *synsem* object is *(locally) o-free* provided it is not (locally) o-bound. Two *synsem* entities are *coindexed* provided their LOCAL | CONTENT | INDEX values are token-identical." (Pollard and Sag, 1994, p. 401)

(10) $\texttt{loc-o-bind}(x, y) \stackrel{\forall}{\Longleftarrow}$
$$ \texttt{loc-o-command}\big({}^{x}[\text{LOC CONT INDEX } \boxed{1}], {}^{y}[\text{LOC CONT INDEX } \boxed{1}] \big) $$

(11) $\texttt{o-bind}(x, y) \stackrel{\forall}{\Longleftarrow}$
$$ \texttt{o-command}\big({}^{x}[\text{LOC CONT INDEX } \boxed{1}], {}^{y}[\text{LOC CONT INDEX } \boxed{1}] \big) $$

(12) *The* Binding Theory *formalized*

Principle A:

$$\forall x \left(\begin{array}{l} \exists y\ \texttt{loc-o-command}(y,\ {}^{x}[\textsc{loc cont}\ \ ana]) \rightarrow \\ \exists z\ \texttt{loc-o-bind}(z, x) \end{array} \right)$$

Principle B:

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

Principle C:

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

Some aspects of the case theory of Przepiórkowski 1999

First of all, we must note a few important differences between the signature of Pollard and Sag 1994 and the signature of Przepiórkowski 1999:

(13)a. *argument* ARGUMENT *synsem*
RAISED *boolean*

b. *category* HEAD *head*
VALENCE *valence*

c. *head* ARG-ST *list(argument)*

d. *valence* SUBJECT *list(argument)*
SPECIFIER *list(argument)*
COMPLEMENTS *list(argument)*

(14) *The* RAISED *Principle (after Przepiórkowski 1999, p. 424)*

$$unembedded\text{-}sign \rightarrow$$

$$\forall\boxed{1}\ \forall\boxed{2}\ \forall\boxed{3}\ \forall\boxed{4}\forall\boxed{5}$$
$$\left(\begin{array}{l} \left(\begin{array}{l} \boxed{1}\big[\textsc{head}\ \ \boxed{2}[\textsc{arg-st}\ \ \boxed{3}]\big] \wedge \\ \texttt{member}\big(\boxed{4}[\textsc{arg}\ \ \boxed{5}], \boxed{3}\big) \end{array} \right) \rightarrow \\ \\ \left(\begin{array}{l} \boxed{4}[\textsc{raised}\ \ +] \ \leftrightarrow \\ \quad \exists\boxed{6}\ \exists\boxed{7}\ \exists\boxed{8}\ \exists\boxed{9} \\ \qquad \left(\begin{array}{l} \boxed{7}[\textsc{arg-st}\ \ \boxed{6}] \wedge \\ \texttt{member}\big(\boxed{8}[\textsc{arg}\ \ \boxed{5}], \boxed{6}\big) \wedge \\ \texttt{member}\big(\boxed{9}[\textsc{arg loc cat head}\ \ \boxed{2}], \boxed{6}\big) \end{array} \right) \end{array} \right) \end{array} \right)$$

Przepiórkowski 1999 assumes that the HEAD values of two *category* entities are identical if and only if one is the CATEGORY value of a syntactic projection of the other. Supposing that this is the case, (14) says that in each utterance, for each *category* component, $\boxed{1}$, with HEAD value $\boxed{2}$, and for each *argument* entity, $\boxed{4}$, on the ARG-ST list of $\boxed{1}$, $\boxed{4}$ is raised if and only if there is a *head* entity, $\boxed{7}$, in the utterance with an ARG-ST list, $\boxed{6}$, that contains an *argument* entity with the *synsem* entity that is also the ARGUMENT value of $\boxed{4}$ as its ARGUMENT value and another *argument* that has the same HEAD value as $\boxed{1}$, i.e., a projection of $\boxed{1}$ is selected by means of the ARG-ST list of $\boxed{7}$.

(15) *A case principle of Polish (after Przepiórkowski 1999, p. 427)*

$$\forall\boxed{1}\ \forall\boxed{2}$$
$$\left(\begin{array}{l} \left(\begin{array}{l} \left[\begin{array}{ll} verbal & \\ \textsc{neg} & + \\ \textsc{arg-st} & \langle object | \boxed{2}\ list \rangle \end{array} \right] \\ \wedge\ \texttt{member}\big(\boxed{1}[\textsc{arg loc cont index}\ \ ref], \boxed{2}\big) \\ \wedge\ \texttt{str-cfc}\big(\boxed{1}\big) \end{array} \right) \rightarrow \\ \boxed{1}[\textsc{arg loc cat head agr case}\ \ sgen] \end{array} \right)$$

Informally (and simplifying a bit), the principle in (15) says that an argument of a negated verb that is not its least oblique argument, has a referential index, and is a "complete functional complex" with structural case receives structural genitive case. A *complete functional complex* is defined as an *argument* entity that is marked as not being raised and whose SUBJ list only contains arguments that are marked as not being raised. The unary relation `str-cfc` holds of exactly those entities in a model that are a complete functional complex with structural case (Przepiórkowski, 1999, p. 425).

Defining the syntax of Ty2 in HPSG

An illustration of the goal:

(16) *The relevant part of the lexical entry of* every

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{every} \rangle \\ \text{SS LOC CONT LF} & \lambda P \lambda Q.\forall x[P(w)(x) \rightarrow Q(w)(x)] \end{bmatrix}$$

NP

$$\begin{bmatrix} \text{PHON} & \langle \textit{every, unicorn} \rangle \\ \text{SS LOC CONT LF} & \lambda Q.\forall x[\mathsf{unicorn}'(w)(x) \rightarrow Q(w)(x)] \end{bmatrix}$$

C: Det

$$\begin{bmatrix} \text{PHON} & \langle \textit{every} \rangle \\ \text{SS LOC CONT LF} & \lambda P \lambda Q.\forall x[P(w)(x) \rightarrow Q(w)(x)] \end{bmatrix}$$

H: N

$$\begin{bmatrix} \text{PHON} & \langle \textit{unicorn} \rangle \\ \text{SS LOC CONT LF} & \mathsf{unicorn}'(w) \end{bmatrix}$$

A reminder: The expressions of Ty2

Definition 1. *The* ***meaningful expressions of Ty2*** *are the smallest family* $(\mathrm{Ty2}_\tau)_{\tau \in Types}$ *such that*

for each $\tau \in Types$, $Var_\tau \cup Const_\tau \subset \mathrm{Ty2}_\tau$,

for each $\tau \in Types$, *for each* $\tau' \in Types$,

if $\alpha \in \mathrm{Ty2}_{\langle \tau', \tau \rangle}$ *and* $\beta \in \mathrm{Ty2}_{\tau'}$, *then* $\alpha(\beta) \in \mathrm{Ty2}_\tau$,

for each $\tau \in Types$, *for each* $\tau' \in Types$, *for each* $n \in \mathbb{N}$, *for each* $v_{n,\tau'} \in Var_{\tau'}$, *for each* $\alpha \in \mathrm{Ty2}_\tau$,

$(\lambda v_{n,\tau'}.\alpha) \in \mathrm{Ty2}_{\langle \tau', \tau \rangle}$,

for each $\tau \in Types$, *for each* $\alpha \in \mathrm{Ty2}_\tau$, *for each* $\beta \in \mathrm{Ty2}_\tau$,

$(\alpha = \beta) \in \mathrm{Ty2}_\tau$,

for each $\alpha \in \mathrm{Ty2}_t$,

$\neg\alpha \in \mathrm{Ty2}_t$,

for each $\alpha \in \mathrm{Ty2}_t$, *for each* $\beta \in \mathrm{Ty2}_t$,

$(\alpha \wedge \beta) \in \mathrm{Ty2}_t$, *and* *(analogously for* $\vee, \rightarrow, \leftrightarrow$*)*

for each $\tau \in Types$, *for each* $n \in \mathbb{N}$, *for each* $v_{n,\tau} \in Var_\tau$, *for each* $\alpha \in \mathrm{Ty2}_t$,

$\exists v_{n,\tau}\alpha \in \mathrm{Ty2}_t$. *(and analogously for* $\forall$*)*

We are now ready to define the RSRL grammar of the meaningful expressions of Ty2. The specification of the grammar of Ty2 closely follows the appendix to Richter and Sailer 1999a with the modifications proposed in Sailer 2000. The ontology of Ty2 expressions is determined by a hierarchy of sorts under the supersort *meaningful-expressions* (*me*). The sort *me* will be defined as appropriate value for the pair $\langle content, \text{LOGICAL-FORM} \rangle$, and thus be anchored in the feature geometry of signs.

(17) The sort hierarchy of the sort *meaningful-expression* (*me*):

- *me* TYPE *type*
  - *var(iable)* N-NUM(BER) *n-num(ber)*
  - *const(ant)* N-NUM(BER) *n-num(ber)*
  - *appl(ication)* FUNC(TOR) *me*, ARG(UMENT) *me*
  - *abstr(action)* VAR *variable*, ARG *me*
  - *equ(ation)* ARG1 *me*, ARG2 *me*
  - *neg(ation)* ARG *me*
  - *l(ogical)-const(ant)* AGR1 *me*, ARG2 *me*
    - *dis(junction)*
    - *con(junction)*
    - *imp(lication)*
    - *bi-imp(lication)*
  - *quant(ifiers)* VAR *variable*, SCOPE *me*
    - *uni(versal)*
    - *exi(stential)*

The attribute TYPE is defined on every entity of sort *me*. Its value encodes the type of the expression and can be read as a subscript of the expression. Entities of sort *var* and entities of sort *const* have one more subscript. The second subscript is the natural number, n, that is used to number the variables (and constants) of each type. To number the variables and constants of each type, a representation of natural numbers is needed. With existential component quantification, a theory of natural numbers can easily be defined on the basis of a simple signature:

(18) The natural numbers:

- *n-number*
  - *zero*
  - *non-zero* N-NUMBER *n-number*

The theory of the natural numbers:

$$n\text{-}number \rightarrow \exists x \ {}^{x}[zero]$$

Every meaningful expression has a subscript that encodes its type. A type is either atomic or a pair. There is a species for each atomic type, *entity* for the type e, *truth* for the type t, and *w-index* for the type s. Complex types are modeled by entities of sort *complex-type*. They

have two attributes, IN and OUT, which are again *type*-valued. The entity under IN models the first type in the pair that constitutes a complex type, and the entity under OUT models the second type in the pair.

(19) The sort hierarchy of *type*:

- *type*
  - *atomic-type*
    - *entity*
    - *truth*
    - *w-index*
  - *complex-type* IN *type*, OUT *type*

(20) Type restrictions on the logical operators:

$$appl \rightarrow \begin{bmatrix} \text{TYPE} & \boxed{2} \\ \text{FUNC TYPE} & \begin{bmatrix} \text{IN} & \boxed{1} \\ \text{OUT} & \boxed{2} \end{bmatrix} \\ \text{ARG TYPE} & \boxed{1} \end{bmatrix},$$

$$abstr \rightarrow \begin{bmatrix} \text{TYPE} & \begin{bmatrix} \text{IN} & \boxed{1} \\ \text{OUT} & \boxed{2} \end{bmatrix} \\ \text{VAR TYPE} & \boxed{1} \\ \text{ARG TYPE} & \boxed{2} \end{bmatrix},$$

$$equ \rightarrow \begin{bmatrix} \text{TYPE} & truth \\ \text{ARG1 TYPE} & \boxed{1} \\ \text{ARG2 TYPE} & \boxed{1} \end{bmatrix},$$

$$neg \rightarrow \begin{bmatrix} \text{TYPE} & truth \\ \text{ARG TYPE} & truth \end{bmatrix},$$

$$l\text{-}const \rightarrow \begin{bmatrix} \text{TYPE} & truth \\ \text{ARG1 TYPE} & truth \\ \text{ARG2 TYPE} & truth \end{bmatrix},$$

$$quant \rightarrow \begin{bmatrix} \text{TYPE} & truth \\ \text{SCOPE TYPE} & truth \end{bmatrix}$$

We must impose two more conditions, and we will add a third one that will ensure that meaningful expressions of Ty2 will have a representation with a unique configuration. Thus we will

- exclude cyclic structures,
- exclude infinite structures, and
- require that in each meaningful expression, isomorphic configurations be identical.

We use a technique introduced by Sailer 2000 for a General Non-Cyclicity Principle. Sailer uses a binary relation, `component`. The appropriate meaning of `component` can be defined in grammars with signatures whose set of attributes is finite and whose set of relation symbols contains a symbol `component` that is assigned arity two by the arity function. Assume that $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$ is such a signature. The set of clauses that define `component` is the smallest set C such that

$\texttt{component}(x,y) \stackrel{\forall}{\Longleftarrow} x = y \ \in C$, and

for each $\alpha \in \mathcal{A}$,

$\texttt{component}(x,y) \stackrel{\forall}{\Longleftarrow} {}^{y}[\alpha\ \boxed{1}] \wedge \texttt{component}(x,\boxed{1}) \ \in C$.

In the definition of the meaning of `component`, x is a component of y.

The Ty2 Non-Cyclicity Principle is formulated with respect to a signature, $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, that includes the signature of Ty2 and has a finite set of attributes.

(21) The Ty2 Non-Cyclicity Principle:

$$ty2 \rightarrow \forall \boxed{1} \left(\left(\bigvee_{\alpha \in \mathcal{A}} [\alpha\ \boxed{1}] \right) \rightarrow \neg\, \texttt{component}(:, \boxed{1}) \right)$$

The consequent of the principle in (21) is Sailer's GNP. The Non-Cyclicity Principle can be paraphrased as saying that for each entity, o, in the grammar of Ty2, if an attribute α is defined on o and $\boxed{1}$ is its value, then o is not a component of $\boxed{1}$. Informally, o is not cyclic.

Similarly, Sailer 2000 shows that it is possible to formulate a General Finiteness Principle (GFP) over signatures with a finite set of attributes. The GFP requires that every entity in every model of the grammar has finitely many components. Again, Sailer's general principle can easily be restricted to some subset of entities in the model of a grammar by turning it into the consequent of a principle with an appropriate antecedent.

(22) The Ty2 Finiteness Principle:

$$ty2 \rightarrow \exists \boxed{1} \forall \boxed{2} (\texttt{component}(\boxed{2}, :) \rightarrow \texttt{member}(\boxed{2}, \boxed{1}[chain]))$$

For the final principle, the Ty2 Identity Principle, we need an auxiliary relation. Following Sailer 2000, we define the meaning of copy in one clause:

(23)
$$\texttt{copy}(x,y) \stackrel{\forall}{\Longleftarrow} \bigvee_{\sigma \in \mathcal{S}} ({}^{x}[\sigma] \wedge {}^{y}[\sigma]) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\forall \boxed{1} ({}^{x}[\alpha\ \boxed{1}] \rightarrow \exists \boxed{2} ({}^{y}[\alpha\ \boxed{2}] \wedge \texttt{copy}(\boxed{1}, \boxed{2}))))$$

y is a copy of x if x and y have the same species (second line), and if an attribute α with value $\boxed{1}$ is defined on x, then α is also defined on y, and $\boxed{1}$ and the value of α on y, $\boxed{2}$, are in the copy relation (third line).

Two entities that stand in the copy relation are of the same species, and they have the same attributes defined on them; and corresponding attribute values stand again in the copy relation. In the Ty2 Identity Principle, we require that all the corresponding entities in the copy relation of representations of Ty2 expressions be identical:

(24) The Ty2 Identity Principle:

$$ty2 \rightarrow \forall \boxed{1} \forall \boxed{2} (\texttt{copy}(\boxed{1}, \boxed{2}) \rightarrow \boxed{1} = \boxed{2})$$

With the Ty2 grammar in place, we can formulate a first approximation to a Semantics Principle (SP).

(25) Semantics Principle (first approximation)

$$phrase \rightarrow \left(\begin{bmatrix} \text{SYNSEM} & \begin{bmatrix} \text{LOC CONTENT LF} & \begin{bmatrix} application & \\ \text{FUNC} & \boxed{3} \\ \text{ARG} & \begin{bmatrix} abstraction & \\ \text{VAR} & \begin{bmatrix} \text{TYPE} & w\text{-}index \\ \text{N-NUM} & zero \end{bmatrix} \\ \text{ARG} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR SYNSEM LOC CONTENT LF} & \boxed{1} \\ \text{NONHEAD-DTR SYNSEM LOC CONTENT LF} & \boxed{2} \end{bmatrix} \end{bmatrix} \wedge \left(\begin{array}{c} (\boxed{1} = \boxed{3} \quad \wedge \quad \boxed{2} = \boxed{4}) \\ \vee \\ (\boxed{1} = \boxed{4} \quad \wedge \quad \boxed{2} = \boxed{3}) \end{array} \right) \right)$$

Sailer 2000 proves the crucial properties of the RSRL grammar of Ty2:

Firstly, there is an exhaustive model of the grammar whose universe contains exactly the natural numbers, the set of types, $Types$, and the meaningful expressions of Ty2. This result shows that the grammar indeed specifies what we want to specify with it.

Secondly, for each indiscernibility class of entities of sort me in an exhaustive model of the grammar, there exists a corresponding Ty2 expression that has the same meaning.

Thirdly, for each Ty2 expression there exists an AVM description that picks out exactly the indiscernibility class of entities in models of the RSRL grammar of Ty2 whose me entities correspond to the Ty2 expression. It is this result that allows us to use Ty2 expressions in AVM descriptions in places where the AVM syntax would require a description of the Ty2 expression.

In HPSG, Ty2 has been used in an analysis of negative polarity items in French (Richter and Sailer, 1999a) and in an analysis of negative concord in Polish (Richter and Sailer, 1999b). Richter and Sailer 1999c uses Ty2 as an example of how an underspecified representation language for an arbitrary semantic representation language can be defined in HPSG. Sailer 2000 is the most comprehensive discussion of Ty2 as semantic representation language in HPSG. It explores different ways of formalizing a grammar of Ty2, adds type shifting rules, and uses the system in an analysis of the syntax and semantics of idiomatic expressions in English.

An operational extension

(26) The sort hierarchies of *content* and of *β-reduction*:

$$
\begin{array}{lll}
content & \textsc{logical-form} & me \\
 & \beta\textsc{-reduction} & \beta\textit{-reduction}
\end{array}
$$

$$
\begin{array}{lll}
\beta\textit{-reduction} & \textsc{term} & me \\
 & \textsc{aux} & me \\
\quad no\textit{-reduction} & & \\
\quad one\textit{-reduction} & \textsc{rec} & \beta\textit{-reduction}
\end{array}
$$

(27) The theory of *β-reduction*:

$$no\text{-}reduction \rightarrow \begin{bmatrix}\textsc{term} & \boxed{1}\\ \textsc{aux} & \boxed{1}\end{bmatrix}$$

$$
\begin{array}{l}
one\text{-}reduction \rightarrow \begin{bmatrix}\textsc{term} & \boxed{1}\\ \textsc{aux} & \boxed{2}\\ \textsc{rec} & \begin{bmatrix}\textsc{term} & \boxed{1}\\ \textsc{aux} & \boxed{3}\end{bmatrix}\end{bmatrix} \\
\quad \land\ \texttt{component}\left({}^{y}\begin{bmatrix} appl & \\ \textsc{func} & \begin{bmatrix} abstr & \\ \textsc{var} & \boxed{5}\\ \textsc{arg} & \boxed{4}\end{bmatrix}\\ \textsc{arg} & \boxed{6}\end{bmatrix}, \boxed{2}\right) \\
\quad \land\ \texttt{replace}(\boxed{4}, \boxed{5}, \boxed{7}, \boxed{6}) \\
\quad \land\ \texttt{replace}(\boxed{2}, y, \boxed{3}, \boxed{7})
\end{array}
$$

(28) The relation `replace`:

The base cases:

$$\texttt{replace}(x, y, v, w) \stackrel{\forall}{\Longleftarrow} x = y \ \land\ v = w$$

$$\texttt{replace}(x, y, v, w) \stackrel{\forall}{\Longleftarrow} \neg x = y \ \land\ {}^{x}[\,var \lor const\,] \land\ x = v$$

An example of a recursive case:

$$
\begin{array}{l}
\texttt{replace}(x, y, v, w) \stackrel{\forall}{\Longleftarrow} \\
\quad \neg x = y \ \land\ {}^{x}\begin{bmatrix} abstr & \\ \textsc{type} & \boxed{1}\\ \textsc{var} & \boxed{2}\\ \textsc{arg} & \boxed{3}\end{bmatrix} \land\ {}^{v}\begin{bmatrix} abstr & \\ \textsc{type} & \boxed{1}\\ \textsc{var} & \boxed{2}\\ \textsc{arg} & \boxed{4}\end{bmatrix} \land \\
\quad \texttt{replace}(\boxed{3}, y, \boxed{4}, w)
\end{array}
$$

Given the theory of the sort *β-reduction*, we only need to define one more auxiliary relation, `redex-free`, before we can state a principle that demands that the LOGICAL-FORM value of each *content* entity be redex free. The definition of `redex-free` in (29) uses the relation `component`, which we defined above.

(29) The relation `redex-free`:

$$\texttt{redex-free}(x) \stackrel{\forall}{\Longleftarrow} \neg \exists y \left({}^{y}\begin{bmatrix} \textit{appl} \\ \text{FUNC} \ \textit{abstr} \end{bmatrix} \wedge \texttt{component}(y,\ {}^{x}[\textit{me}]) \right)$$

(30) The β-REDUCTION PRINCIPLE:

$$\textit{content} \rightarrow \left(\begin{bmatrix} \text{LOGICAL-FORM} & \boxed{1} \\ \beta\text{-REDUCTION TERM} & \boxed{1} \end{bmatrix} \wedge \texttt{redex-free}(\boxed{1}) \right)$$

(31) *The* SEMANTICS PRINCIPLE *(final version)*

$$\textit{phrase} \rightarrow$$

$$\begin{bmatrix} \text{SS} & \begin{bmatrix} \text{LOC CONT } \beta\text{-REDUCT} & \begin{bmatrix} \beta\textit{-reduction} & \\ \text{AUX} & \begin{bmatrix} \textit{application} & \\ \text{FUNC} & \boxed{3} \\ \text{ARG} & \begin{bmatrix} \textit{abstraction} & \\ \text{VAR} & \begin{bmatrix} \text{TYPE} & \textit{w-index} \\ \text{N-NUM} & \textit{zero} \end{bmatrix} \\ \text{ARG} & \boxed{4} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR SS LOC CONT LF} & \boxed{1} \\ \text{NONHEAD-DTR SS LOC CONT LF} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

$$\wedge \left((\boxed{1} = \boxed{3} \wedge \boxed{2} = \boxed{4}) \vee (\boxed{1} = \boxed{4} \wedge \boxed{2} = \boxed{3}) \right)$$

References

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

Przepiórkowski, Adam 1999. *Case Assignment and the Complement-Adjunct Dichotomy: A Non-Configurational Constraint-Based Approach.* PhD thesis, Eberhard-Karls-Universität Tübingen.

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

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*, (= *Arbeitspapiere des SFB 340, Nr. 132, Volume 1*), 231–300. Eberhard-Karls-Universität Tübingen.

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 HPSG*, 247–282. CSLI Publications.

Richter, Frank and Sailer, Manfred 1999c. Underspecified Semantics in HPSG. In Harry Bunt and Reinhard Muskens (eds), *Computing Meaning*, (= *Studies in Linguistics and Philosophy*), 95–112. Kluwer academic publishers.

Sailer, Manfred 2000. *Combinatorial Semantics and Idiomatic Expressions in Head-Driven Phrase Structure Grammar. Submitted Version of June 29th, 2000*. PhD thesis, Eberhard-Karls-Universität Tübingen.