Underspecified Semantics in HPSG

Frank Richter Manfred Sailer

Universität Tübingen
Germany
email: {fr,mf}@sfs.nphil.uni-tuebingen.de

Abstract

In this paper we propose a new approach to underspecified semantics in HPSG. We show that a general framework for underspecified semantic representation languages can be reconciled with the traditional logical architecture of HPSG as outlined in (Pollard and Sag 1994) and formalized in (King 1994). As an example of a semantic object language that can be treated within this scenario, an extensional typed logic with lambda abstraction is extended to an underspecified representation language as described in (Bos 1995), and the resulting language is modeled in HPSG.

Keywords: underspecification, HPSG, hole semantics, scope ambiguities

1 Introduction

Theories of semantic underspecification assign semantically ambiguous sentences a unique semantic representation. In computational applications of natural language processing, the need for a compact representation arises in order to avoid potentially highly disjunctive specifications of different readings which scopal ambiguities may induce. Typical instances of ambiguities treated by underspecification are scope ambiguities of adjectives in NPs with respect to PPs as illustrated in (1a), and different possible scopings of a quantifier with respect to some other scope bearing element in sentences like (1b) (cf. (Pinkal 1995), (Reyle 1993)).

(1) a. a former professor in Tübingen
`former(in-Tübingen(professor))` `in-Tübingen(former(professor))`

b. every professor didn't answer
`every-professor(not(answer))` `not(every-professor(answer))`

While HPSG is frequently used for the implementation of natural language fragments ((Hinrichs, Meurers and Nakazawa 1994), (Keller 1994), (Frank 1994)), the main focus of research in the HPSG framework has been on syntactic phenomena; and the semantic concepts traditionally appealed to have been borrowed from situation semantics (Pollard and

Sag 1987), (Pollard and Sag 1994). Since HPSG is widely used as a basis for computational implementations, the question arises if a general framework for semantic underspecification can be integrated with it. Such a framework is presented in (Bos 1995), who defines a general formalism for a semantic representation language that can be applied to semantic object languages such as predicate logic, dynamic predicate logic and discourse representation theory. If the envisaged integration is possible, a number of semantic theories would become available to HPSGians.

In the following sections, we show how Bos's semantic representation language can be defined in HPSG. As a concrete example of an object language, we choose Ty2 (Gallin 1975), which is one of the standard formal languages in natural language semantics.

2 Syntax and semantics in HPSG

According to (Pollard and Sag 1994) and (King 1994) a grammar in the HPSG formalism consists of a signature and a theory which is a set of descriptions. In the signature, the sort hierarchy, attribute names and appropriateness function are defined. For example, in the grammar presented in (Pollard and Sag 1994), *verb* is a maximally specific subsort of *head*, VFORM is an attribute, and the appropriateness function on $\langle$*head*, VFORM$\rangle$ yields the sort *vform* as its value. The maximally specific sorts in the sort hierarchy are sometimes called species. The signature fixes the ontology of the linguistic universe and generates a set of descriptions which is closed under negation, conjunction and disjunction. We call the descriptions which comprise a theory the constraints of that theory. Usually they are stated as implications. Prominent examples are the Head Feature Principle and the Subcategorization Principle. Every object in the linguistic universe must satisfy every description in the theory, and the linguistically intended model contains instances of all objects that are permitted by the theory. In a sense, the intended model is the largest possible model of the grammar. When we talk about objects denoted by the grammar, we implicitly refer to objects in such an exhaustive model.

Given that the HPSG formalism uses a logic of descriptions, a natural place for underspecification is the logical description language. Whereas objects in the model of the grammar must be totally well-typed and sort resolved, i.e. of a maximally specific sort containing all possible sub-objects, descriptions are usually partial. Typical instances of descriptive underspecification are the use of sorts which are not maximally specific and the use of relations. Sorts that are not maximally specific denote the set of objects whose species is below them in the sort hierarchy, and there may be several n-tuples which satisfy an n-ary relation.

In HPSG, syntax and semantics are treated on the same level of representation, and both are subject to the same set of constraints. (Nerbonne 1992) and (Frank and Reyle 1995) exploit the uniform logical architecture of the grammar for a treatment of scope ambiguities in the description language. The idea is to specify the grammar in such a way that, while there is exactly one syntactic structure for a given sign, there can be multiple semantic structures which each correspond to one reading of the sentence. One should notice that, in this approach, every concrete object in the denotation of the grammar has a semantic structure which unambiguously determines a unique reading for that object.

Our own approach to semantic underspecification differs from those described in two respects. First, we provide a general framework which allows for the integration of several logical languages as underspecified semantic representation languages. (Nerbonne 1992) and (Frank and Reyle 1995) are only concerned with one logical language, namely a language for generalized quantifiers and DRT, respectively. Second, we do not locate semantic underspecification in the description language. Rather, a sentence with multiple readings receives a unique syntactic and a unique semantic structure. In other words, there is only one object in the grammar that represents this sentence. This, in turn, means that the semantic representation itself is vague, and its interpretation is a set of readings rather than a single reading.

One advantage of this approach is that it is possible to draw on the literature on underspecified semantic representation languages (c.f. (Pinkal 1995), (Bos 1995), (Reyle 1993) among others). Second, in a computational application, the number of HPSG structures to be computed for ambiguous sentences is reduced to one. For this idea to work, it is essential that sentences subject to semantic underspecification indeed have a single syntactic structure. In traditional approaches this is usually not the case. For example, in most theories, (1a) above receives distinct syntactic analyses for the two readings given. In HPSG, the clear differentiation between phonological word order (phenogrammatical structure) and syntactic structure (tectogrammatical structure) in linearization grammars provides the theoretical background for a syntactic analysis that is essentially underspecified (Kathol 1995). As we will see in example (18) below, in a linearization grammar, the syntactic tree position of adjuncts and complements can be fixed independently of word order. Constraints that take the actual phonological word order into account can then be used to derive a semantic representation appropriate for a given syntactic structure and word order. If word order permits, we obtain an underspecified semantic structure.

In the following sections, we restructure the CONTENT value of HPSG signs. It will be shown that our new *content* objects can be mapped to an underspecified semantic representation language as defined in (Bos 1995). We are able, thus, to establish a direct link between grammars in HPSG and a general metalanguage for semantic underspecification.

3 An unplugged semantics for HPSG

In this section, we briefly summarize the proposal made in (Bos 1995) for an underspecified semantic representation. We then show that this proposal can be cast in terms of an HPSG signature. Finally, a mapping from the objects in the model of an HPSG theory to underspecified semantic representations is defined.

Bos (1995) defines an underspecified semantic representation in the following way. Given some formal language, $\mathcal{L}$, a set of *labels*, and a set of metavariables called *holes*, $\mathcal{L}$ is extended to a language, $\mathcal{L}U$, which allows for metavariables in its syntax. A *labeled formula* is an element of $\mathcal{L}U$ together with a label. A *constraint* is of the form $k \leq k'$, where k, k' are taken from the union of the set of holes and the set of labels. An *underspecified representation* (UR) is a triple $\langle H, L, C\rangle$ that consists of a set of holes, a set of labeled formulae and a set of constraints.

Bos (1995) is only interested in certain kinds of URs, namely URs that have a top element under $\leq$ with respect to a bijection from H to L. Such a bijection is called a *possible plugging.* The interpretation of a UR is the set of interpretations of the top-labeled formula of L under each possible plugging.

For our purposes, we do not introduce labeled formulae. Instead, we use formulae of $\mathcal{L}U$ directly. To stay close to Bos's terminology, however, we will sometimes refer to formulae of $\mathcal{L}U$ as labels. In (2), we formally define some notions that we need later.

(2) a. Let $Hole$ be the set of metavariables.

b. Let $Meta$ be the set of all formulae of $\mathcal{L}U$ (each element of $Hole$ is in $Meta$). For each $H \subseteq Hole$, $Meta_H$ is the subset of $Meta \setminus Hole$ that contains no metavariables which are not in H.

c. Let $Constr$ be the set of constraints such that for each $k, k' \in Meta$, $k \leq k' \in Constr$. For each $M \subseteq Meta$, $Constr_M$ is the set of constraints that only contains formulae from M.

d. Let $\mathcal{UR}$ be the set of all triples $\langle H, L, C \rangle$ such that $H \subseteq Hole$, $L \subseteq Meta_H$, $C \subseteq Constr_{H \cup L}$.[1]

The semantics of elements of $\mathcal{UR}$ is defined exactly as in (Bos 1995) and as described above.

Having defined the semantic representation we are aiming for, we can now define an HPSG signature and a mapping, SR, from objects licensed by that signature to the union of the sets defined in (2). For the moment, we ignore those parts of the definitions of SR that depend on peculiar characteristics of $\mathcal{L}$. In the next section, a specific formal language will be substituted for $\mathcal{L}$, and the definition of SR will be completed.

We add a sort, *meta*, with subsorts *hole* and *label* to the HPSG sort hierarchy. SR maps each finite, acyclic object of sort *meta* to an element of $Meta$ in such such way that each *hole* object is mapped to an element of $Hole$, and each *label* object is mapped to a formula of the language, $\mathcal{L}U$.

Secondly, we introduce a sort, *constr(aint)*, in order to capture the subordination constraints of (Bos 1995). The attributes A1 and A2 are appropriate to *constr* objects and yield objects of sort *meta* as values. The representation function, SR, maps each finite, acyclic object of the sort *constr* to a member of $Constr$, such that, for each such object, u, of sort *constr* with u_1 as its A1 value and u_2 as its A2 value, $SR(u) = SR(u_1) \leq SR(u_2)$.

Finally, we introduce a sort, *underspecified-representation* (*ur*), which corresponds to Bos's URs. Three list-valued attributes, HOLES, LABELS and CONSTRAINTS, are appropriate to *ur*. The values of these attributes correspond to the sets H, L and C, respectively. In addition, we use the auxiliary attributes MAIN and TOP to make global information

[1] In our HPSG formalization, lists are used to represent the sets of holes, metavariables and constraints. Since, in our semantic representations, we are only concerned with finite sets, nothing hinges on this difference.

locally available.[2] *ur* objects replace the traditional *content* objects of HPSG. The appropriateness conditions for the sort *underspecified-representation* are given in (3).

(3) *u(nderspecified-)r(epresentation)*

	HOLES	*list(hole)*
	LABELS	*list(label)*
	CONSTR(AINTS)	*list(constr)*
	MAIN	*label*
	TOP	*hole*

For the definition of SR on *ur* objects, an auxiliary representation function, LR, from *list* objects to lists is needed:

For each finite, acyclic object u of sort *list*,
if u is of sort *elist*, then $LR(u) = \langle\ \rangle$,
if u is of sort *nelist* with u_1 as its FIRST value and u_2 as its REST value, then

$$LR(u) = \langle SR(u_1) | LR(u_2) \rangle.$$

Using LR, we let SR map each finite, acyclic object of sort *ur* to a member of $\mathcal{UR}$ such that, for each object u of sort *ur*, with h as its HOLES value, l as its LABELS value and c as its CONSTR value,

$$SR(u) = \langle LR(h), LR(l), LR(c) \rangle.$$

For each *ur* object u, $SR(u) \in \mathcal{UR}$. The linguistic theory must ensure that each theoretically relevant *sign* object in the model[3] has a finite, acyclic CONTENT value, u.

4 Ty2 as an unplugged semantic language

In this section, a concrete semantic object language is substituted for $\mathcal{L}$. As an example of an unplugged language, $\mathcal{L}U$, we adopt a version of the language Ty2 as presented in (Groenendijk and Stokhof 1982). Ty2 is an extensional typed logic with lambda abstraction. We write $Type$ for the set of types which can be constructed from the basic types e, t, s. We extend Ty2 to include holes, i.e. metavariables. The extended language will be referred to as Ty2U. In (4) the set of well-formed expressions of Ty2U is defined.

(4) For each set Var of variable symbols, for each set $Const$ of constant symbols, and for each set $Hole$ of metavariable symbols, Ty2U is the smallest set such that,

for each $x \in Var \cup Const \cup Hole$, and for each $\tau \in Type$, $x_\tau \in$ Ty2U,

for each $\phi_{\langle\tau,\tau'\rangle}, \psi_\tau \in$ Ty2U,$(\phi_{\langle\tau,\tau'\rangle}\psi_\tau)_{\tau'} \in$ Ty2U,

for each $v \in Var$, for each $\tau \in Type$, and for each $\phi_{\tau'} \in$ Ty2U, $(\lambda v_\tau.\phi_{\tau'})_{\langle\tau,\tau'\rangle} \in$ Ty2U,

for each $\phi_t, \psi_t \in$ Ty2U,$(\neg\phi_t)_t, (\phi_t \wedge \psi_t)_t, (\phi_t \vee \psi_t)_t, (\phi_t \rightarrow \psi_t)_t \in$ Ty2U,

for each $v \in Var$, for each $\tau \in Type$, and for each $\phi_t \in$ Ty2U,$(\exists v_\tau.\phi_t)_t$, $(\forall v_\tau.\phi_t)_t \in$ Ty2U.

[2]Compare with the use of the relations **main** and **top** in (Bos, Gambäck, Lieske, Mori, Pinkal and Worm 1996).

[3]In the present framework, saturated phrases which only have a finite number of constituents are theoretically relevant signs.

We assume that the set of constants, $Const$, is finite, and each constant symbol is assigned a specific semantic type. For this purpose, we use a function, $\mathcal{C}$, from $Const$ to $Type$. We then define $\mathrm{Ty2U}_{\mathcal{C}}$ as that subset of Ty2U in which, for each $c \in Const$, each occurrence of an expression of the form c_τ is such that $\tau = \mathcal{C}(c)$.

As Ty2U is a typed language, we must include semantic types in the HPSG signature. This is done with the sort hierarchy in (5).

(5) Sort hierarchy for semantic types:

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

We can now refine the sort hierarchy of *meta* given in the last section. The attribute TYPE is defined on *meta*. *hole* is a species; and the species below *label* mimic variables, constants, application, abstraction, logical constants and quantifiers. In (6), the sort hierarchy below *meta* is expressed by identation.

(6) The sort hierarchy below *meta*:

```
meta   TYPE  type
       hole
       label
           var(iable)
           const(ant)
               const_1
               ...
               const_n
           appl(ication)    FUNC(TOR)     meta
                            ARG(UMENT)    meta
           abstr(action)    VAR   variable
                            ARG   meta
           neg(ation)   ARG   meta
           l(ogical)-const(ant)    AGR1   meta
                                   ARG2   meta
               dis(junction)
               con(junction)
               imp(lication)
           quant(ifiers)    VAR     variable
                            SCOPE   meta
               universal
               existential
```

The following descriptions constrain the logical types of the subsorts of *meta* in the usual way:[4]

[4] In (7), we use the syntax of King (1994). Attributes are written in small caps; sorts, in italics. A path consists of the colon ':' and a sequence of attributes. $\pi \sim \sigma$ means that path π is defined, and has a value of sort σ. $\pi \approx \pi'$ means that paths π and π' are defined and have identical values. The logical constants, $\wedge$, $\vee$, $\rightarrow$ and $\neg$, receive a classical set theoretic interpretation.

(7) Type restrictions:

:∼*appl* → :FUNC TYPE IN ≈ :ARG TYPE ∧ :FUNC TYPE OUT ≈ :TYPE

:∼*abstr* → :TYPE IN ≈ :VAR TYPE ∧ :TYPE OUT ≈ :ARG TYPE

:∼*neg* → :ARG TYPE∼*truth* ∧ :TYPE∼*truth*

:∼*l-const* → :ARG1 TYPE∼*truth* ∧ :ARG2 TYPE∼*truth* ∧ :TYPE∼*truth*

:∼*quant* → :SCOPE TYPE∼*truth* ∧ :TYPE∼*truth*

For each species below *const*, there is a constraint that fixes its semantic type. In (8), such a constraint is given for a non-logical constant species of type $\langle s, \langle e, t\rangle\rangle$, like *sleep*, which we assume to be the species $const_{127}$.

(8) :∼$const_{127}$ → :TYPE IN∼*w-index* ∧ :TYPE OUT IN∼*entity* ∧ :TYPE OUT OUT∼*truth*

We can now give the final definitions of the function SR. The definition in (9) treats the mapping of *type* objects into $Type$. This mapping is needed in (10) to state the mapping from the set of *meta* objects into the set of Ty2U expressions.

(9) For each finite, acyclic object u of sort *type*:

if u is of sort *entity*, then $SR(u) = e$,

if u is of sort *truth*, then $SR(u) = t$,

if u is of sort *w-index*, then $SR(u) = s$,

if u is of sort *c-type*, with u_1 as its IN value and u_2 as its OUT value, then $SR(u) = \langle SR(u_1), SR(u_2)\rangle$.

When we map a *meta* object to a Ty2U expression, we must take care of the variables, metavariables and constants which occur in Ty2U expressions. For this purpose, we parameterize SR with respect to three functions, V, H and C.

V is a bijection from the set of *var* objects to $Var \times Type$, such that, for each object u of sort *var*, with u' as its TYPE value, if $V(u) = v_\tau$, then $SR(u') = \tau$. H is a bijection from the set of *hole* objects to $Hole \times Type$, such that, for each object u of sort *hole*, with u' as its TYPE value, if $V(u) = h_\tau$, then $SR(u') = \tau$. C is a bijection from the set of species below *const* to $Const \times Type$, such that, for each species c_i below *const*, and for each u of sort c_i with u' as its TYPE value, if $C(u) = c_\tau$, then $SR(u') = \tau = \mathcal{C}(c)$.

(10) Given the functions $\mathcal{C}, V, H, C$, for each object u of sort *meta* with TYPE value t:

if u is of sort *hole*, then $SR(u) = H(u)$,

if u is of sort *var*, then $SR(u) = V(u)$,

for each c_i that is a species below *const*, if u is of sort c_i, then $SR(u) = C(u)$,

if u is of sort *appl* with u_1 as its FUNC value and u_2 as its ARG value, then $SR(u) = (SR(u_1)SR(u_2))_{SR(t)}$,

and analogously for the other subsorts of *label*.

With definition (10), the mapping from the semantic representation of an HPSG sign, i.e. from the *ur* object under its CONTENT attribute to Ty2U, is complete. One should note the following interesting correspondences between *ur* objects and elements of $\mathcal{UR}$, and between elements of $\mathcal{UR}$ and descriptions of *ur* objects:

- The equivalence class of *ur* objects whose SR values are identical can be characterized purely in terms of properties inherent to these objects.
- For each UR, there exists an HPSG description which denotes exactly those *ur* objects that are mapped to UR by SR under some bijections V and H.

The following schema illustrates how the semantic interpretation proceeds.

$$\text{Descr}_{\text{HPSG}} \overset{D}{\Longrightarrow} [\text{ur-object}_{\text{HPSG}}] \overset{SR}{\Longrightarrow} \text{Ty2U} \overset{\text{(Bos 1995)}}{\Longrightarrow} \text{Readings}$$

We augmented an HPSG grammar, Γ, with a signature and descriptions that concern the shape of the CONTENT value of signs. Under HPSG's interpretation function, D (King 1994), the constraints of Γ pertaining to the semantics of signs denote a set of *ur* objects in the model of Γ. Every *ur* object of the equivalence class, [ur-object$_{\text{HPSG}}$], is mapped to the same Ty2U expression. This is guaranteed by the definitions of SR above. Since the semantic representation of HPSG signs is now mapped to Ty2U expressions, we can rely on the definitions of (Bos 1995) for the correct interpretation of these formulae.

In the next section, we illustrate our treatment of semantic underspecification with two linguistic examples.

5 URs in HPSG semantics

In this section, we consider the German translations of the examples given in the introduction. In (11), example (1a) is repeated in German. The readings of the NP in (11) are given below the example. Syntactically, we assume a single representation of the NP, where — independent of the reading — the PP is realized higher up in the tree than the rest of the NP.[5]

(11) (jeder) ehemalige Professor in Tübingen
`former(in-Tübingen(professor))` `in-Tübingen(former(professor))`

In (12), we give the semantic part of the lexical entries used in (11). As we know that each Ty2U expression can be translated into an HPSG description, we can use formulae of Ty2U as a shorthand notation inside AVMs. We use the non-logical constant $\texttt{prec}'_{\langle s,\langle s,t\rangle\rangle}$ which holds between two indices, w_1 and w_2, if w_1 precedes w_2 in time.[6]

[5] See (Richter and Sailer 1996) for a proposal that satisfies this requirement.

[6] By convention, w_i will always stand for a variable of type s, x_i for a variable of type e, and h_i for a hole of any type.

(12) *in Tübingen*: $\begin{bmatrix} \textit{ur} & \\ \text{HOLES} & \langle h_0, h_1 \rangle \\ \text{LABELS} & \left\langle \boxed{2} \lambda w_2 \lambda x_2 . h_1(w_2)(x_2) \wedge \texttt{in-Tübingen}'(w_2)(x_2) \right\rangle \\ \text{CONSTR} & \langle\ \rangle \\ \text{MAIN} & \boxed{2} \\ \text{TOP} & h_0 \end{bmatrix}$

Professor: $\begin{bmatrix} \textit{ur} & \\ \text{HOLES} & \langle h_0 \rangle \\ \text{LABELS} & \left\langle \boxed{1} \lambda w_1 \lambda x_1 . \texttt{prof}'(w_1)(x_1) \right\rangle \\ \text{CONSTR} & \left\langle \boxed{1} \leq h_0 \right\rangle \\ \text{MAIN} & \boxed{1} \\ \text{TOP} & h_0 \end{bmatrix}$

ehemalige: $\begin{bmatrix} \textit{ur} & \\ \text{HOLES} & \langle h_0, h_2 \rangle \\ \text{LABELS} & \left\langle \boxed{3} \lambda w_3 \lambda x_3 . \exists w_4 [\texttt{prec}'(w_4, w_3) \wedge h_2(w_4)(x_3)] \right\rangle \\ \text{CONSTR} & \langle\ \rangle \\ \text{MAIN} & \boxed{3} \\ \text{TOP} & h_0 \end{bmatrix}$

With these lexical entries at hand, we can define a SEMANTICS PRINCIPLE (SP) which constrains the CONTENT value of a phrase according to the CONTENT values of the daughters and the kind of construction. The parts of the SP needed for the examples in this section are given in (13).

(13) SEMANTICS PRINCIPLE: In each phrase:

1. the values of the HOLES and the LABELS attributes are the concatenation of their respective values on the daughters,
2. the value of the MAIN attribute is identical to its value on the head daughter.
3. the value of the TOP attribute is identical to its value on both daughters.
4. the value of the CONSTR attribute is a list, l, which contains exactly the following elements:
   (a) each member of the CONSTR values of the daughters is in l,
   (b) in a *head-adjunct-phrase*, where x is the MAIN value of the nonhead daughter, t is the TOP value and m the MAIN value of the head daughter, $x \leq t$ and $m \leq x$ are in l,
   (c) in a *head-complement-phrase* with a verbal head and quantified NP as complement, where m and t are the MAIN and TOP values of the head, q is the MAIN value of the nonhead, and n is the hole that corresponds to the nucleus of the nonhead, $q \leq t$ and $m \leq n$ are in l,
   (d) in a *phrase* with a quantifier as head and a nominal projection as nonhead, where r is the hole that corresponds to the restriction of the quantifier, and m is the MAIN value of the nominal projection, $m \leq r$ is in l.

Using the lexical entries in (12) and the SP in (13), we obtain a CONTENT value for the NP in (11), which can be described by (14).

(14) $$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{ur}} \\ \text{HOLES} & \langle h_0, h_1, h_2 \rangle \\ \text{LABELS} & \left\langle \boxed{1}, \boxed{2}, \boxed{3} \right\rangle \\ \text{CONSTR} & \left\langle \boxed{1} \leq h_0, \boxed{1} \leq \boxed{2}, \boxed{1} \leq \boxed{3}, \boxed{2} \leq h_0, \boxed{3} \leq h_0 \right\rangle \\ \text{MAIN} & \boxed{1} \\ \text{TOP} & h_0 \end{array}\right]$$

The possible readings of the *ur* object described in (14) are expressed by the possible pluggings of its semantic representation. There are exactly two such pluggings:

(15) a. $P_1 = \{SR(h_0) = SR(\boxed{2}), SR(h_1) = SR(\boxed{3}), SR(h_2) = SR(\boxed{1})\}$

`in-Tübingen(former(professor))`

b. $P_2 = \{SR(h_0) = SR(\boxed{3}), SR(h_1) = SR(\boxed{1}), SR(h_2) = SR(\boxed{2})\}$

`former(in-Tübingen(professor))`

We can now look at an example which involves interaction between a quantifier and an adjunct that has scope over the finite verb, like the sentence in (1b). In (16) the German translations of the two readings are given.

(16) a. daß nicht jeder Professor antwortet

b. daß jeder Professor nicht antwortet

The semantic parts of the relevant lexical entries are given in (17). In addition, we use the lexical entry for *Professor* from (12).

(17) jeder: $$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{ur}} \\ \text{HOLES} & \langle h_0, h_5, h_6, h_7 \rangle \\ \text{LABELS} & \left\langle \boxed{6}\lambda w_3 \forall x_2 . h_5(w_3)(x_2) \rightarrow h_6(w_3), \boxed{7}\lambda w_4 . h_7(w_4)(x_2) \right\rangle \\ \text{CONSTR} & \left\langle \boxed{7} \leq h_6 \right\rangle \\ \text{MAIN} & \boxed{6} \\ \text{TOP} & h_0 \end{array}\right]$$

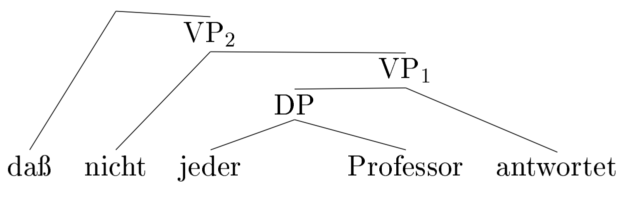
antwortet:

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{ur}} \\ \text{HOLES} & \langle h_0, h_4 \rangle \\ \text{LABELS} & \left\langle \boxed{4}\exists w_1 . h_4(w_1), \boxed{5}\lambda w_2 \lambda x_1 . \texttt{antwort}'(w_2)(x_1) \right\rangle \\ \text{CONSTR} & \left\langle h_0 \leq \boxed{4}, \boxed{4} \leq h_0, \boxed{5} \leq \boxed{4} \right\rangle \\ \text{MAIN} & \boxed{5} \\ \text{TOP} & h_0 \end{array}\right]$$

nicht:

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{ur}} \\ \text{HOLES} & \langle h_0, h_8 \rangle \\ \text{LABELS} & \left\langle \boxed{8}\lambda w_5 . \neg h_8(w_5) \right\rangle \\ \text{CONSTR} & \langle\ \rangle \\ \text{MAIN} & \boxed{8} \\ \text{TOP} & h_0 \end{array}\right]$$

(18)



Applying the Semantics Principle to the syntactic tree described in (18),[7] the following constraints must be on the CONSTR list: At the DP-node, case (4d) applies: h_5 is the restriction hole of the quantifier; and $\boxed{1}$ is the MAIN value of the noun; thus $\boxed{1} \leq h_5$ is on the CONSTR list.

At the VP_1 node, case (4c) applies: $\boxed{5}$ is the MAIN value of the head; and h_0 is its TOP value. $\boxed{6}$ is the MAIN value of the complement; and h_6 is the nucleus hole of the complement; thus $\boxed{5} \leq h_6$ and $\boxed{6} \leq h_0$ are on the CONSTR list.

At the VP_2 node, case (4b) applies: $\boxed{5}$ is the MAIN value of the head; and h_0 is its TOP value. $\boxed{8}$ is the MAIN value of the adjunct. $\boxed{5} \leq \boxed{8}$ and $\boxed{8} \leq h_0$ are, thus, on the CONSTR list.

In addition, the lexical entries contribute the following constraints: $\boxed{7} \leq h_6$ (*jeder*), $\boxed{1} \leq h_0$ (*Professor*), $h_0 \leq \boxed{4}$, $\boxed{4} \leq h_0$, and $\boxed{5} \leq \boxed{4}$ (*antwortet*).

Due to the constraints introduced by the lexical entries and by the SP, there are only two possible pluggings for sentence (18). They are given in (19).

(19) a. $P_1 = \{SR(h_0) = SR(\boxed{4}), SR(h_4) = SR(\boxed{8}), SR(h_5) = SR(\boxed{1}),$
$SR(h_6) = SR(\boxed{7}), SR(h_7) = SR(\boxed{5}), SR(h_8) = SR(\boxed{6})\}$

$\exists w.\neg(\forall x[\texttt{professor}'(w)(x) \to \texttt{antwort}'(w)(x)])$

b. $P_2 = \{SR(h_0) = SR(\boxed{4}), SR(h_4) = SR(\boxed{6}), SR(h_5) = SR(\boxed{1}),$
$SR(h_6) = SR(\boxed{8}), SR(h_7) = SR(\boxed{5}), SR(h_8) = SR(\boxed{7})\}$

$\exists w \forall x.[\texttt{professor}'(w)(x) \to \neg(\texttt{antwort}'(w)(x))]$

It has been observed (Jacobs (1988), Beck (1996)) that, in German, the relative order of a quantified NP and a negation particle determines their relative scope. The two possible word orders for the syntactic structure in (18) thus determine the reading. Sentence (16a) therefore only has the reading implied by plugging P_1; (16b) has the reading expressed by P_2. To capture the relevant generalization, we add the following constraint to part 4 of the SP.

(20) 4. (e) in a sentence with MAIN value m and PHON value P, for each two dominated scope bearing constituents C_1 and C_2 with MAIN values c_1 and c_2 and PHON values P_1 and P_2, if there are *meta* objects m_1 and m_2 in c_1 and c_2 such that $m \leq m_1$ and $m \leq m_2$ are in l, and P_1 precedes P_2 in P, then $c_2 \leq m_1$ is in l.

Adding (20) to the SP in (13) will give us the desired result for the sentences in (16). In these sentences, the negation, *nicht*, as well as the quantifier, *jeder*, are to be considered as 'scope bearing' in the sense of (4e) of the SP. The MAIN value of the sentence is $\boxed{4}$. In the CONSTR list of the sentence, there are, *inter alia*, $\boxed{4} \leq \boxed{8}$, where $\boxed{8}$ is the MAIN value of *nicht*, and $\boxed{4} \leq h_6$, where h_6 is a component of $\boxed{6}$, which is the MAIN value of *jeder*. In

[7] The reader should bear in mind that a linearization grammar is presupposed. The description in (18), therefore, does not determine the phonological, linear precedence relation between *nicht* and *jeder Professor*, i.e. (18) describes both sentences in (16).

(16a), *nicht* phonologically precedes *jeder*, thus, $\boxed{6} \leq \boxed{8}$ is part of the overall CONSTR list. In (16b), *jeder* phonologically precedes *nicht*. (4e) enforces the constraint $\boxed{8} \leq h_6$ on the CONSTR list. It can be confirmed that only P_1 is compatible with the constraint $\boxed{6} \leq \boxed{8}$, and that only P_2 is compatible with the constraint $\boxed{8} \leq h_6$.

For languages like English, where word order is not as directly linked to scope as in German, (20) would not be part of the SP. Thus the corresponding English sentence, (1b), is, in fact, ambiguous.

The two examples considered in this section illustrate that the proposed semantics for HPSG can treat ambiguities for nominal modifiers, quantifiers, and scope bearing adverbials. Word order can furthermore be used to constrain the number of possible readings.

6 Conclusion

Adopting the formalization of HPSG in (King 1994), we argued that the appropriate location for semantic representations in HPSG is in the denotation of the grammar rather than at the level of descriptions. We summarized the work of Bos (1995), who proposes a general framework for extending different object languages to semantic metalanguages, and showed that his approach can be captured in HPSG. For this purpose, we replaced the traditional CONTENT value of HPSG by *underspecified-representation* objects which can be mapped to expressions of a semantic metalanguage constructed from Ty2. Finally, we showed that ambiguities arising from quantifiers and nominal modifiers can be treated in the proposed underspecified HPSG semantics.

The suggested treatment of semantic underspecification offers an interesting perspective of the unified account of syntax and semantics in HPSG. Since, in principle, HPSG is computationally tractable (cf. (Götz and Meurers 1995)), an integral computational treatment of syntax and semantic representation should be possible.

Bos (1995) shows that his treatment of underspecification can be applied to dynamic predicate logic and DRT. We expect that these logics can be substituted for Ty2 in our approach to underspecified representations for HPSG. In this respect, the present proposal is more general than other approaches to underspecified semantics in HPSG (e.g. (Frank and Reyle 1995), (Nerbonne 1992)), which presuppose a specific semantic object language.

References

Beck, S.: 1996, *Wh-constructions and transparent Logical Form*, PhD thesis, Universität Tübingen.

Bos, J.: 1995, Predicate logic unplugged, *Proceedings of the 10th Amsterdam Colloquium.*

Bos, J., Gambäck, B., Lieske, C., Mori, Y., Pinkal, M. and Worm, K.: 1996, Compositional semantics in Verbmobil, *Proceedings of COLING '96.*

Frank, A.: 1994, Verb Second by Lexical Rule or by Underspecification, *Arbeitsberichte des SFB 340 43*, Universität Stuttgart.

Frank, A. and Reyle, U.: 1995, Principle based semantics for HPSG, *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 9 – 16.

Gallin, D.: 1975, *Intensional and Higher order Modal Logic*, North Holland, Amsterdam.

Götz, T. and Meurers, D.: 1995, Compiling HPSG type constraints into definite clause programs, *ACL '95*.

Groenendijk, J. and Stokhof, M.: 1982, Semantic analysis of wh-complements, *Linguistics and Philosophy* **5**, 175 –233.

Hinrichs, E., Meurers, D. and Nakazawa, T.: 1994, Partial-VP and Split-NP Topicalization in German — An HPSG Analysis and its Implementation, *Arbeitspapiere des SFB 340 58*, Universität Tübingen.

Jacobs, J.: 1988, Probleme der freien Wortstellung im Deutschen, *Sprache und Pragmatik* **5**, 8–37.

Kathol, A.: 1995, *Linearization-Based German Syntax*, PhD thesis, Ohio State University.

Keller, F.: 1994, Extraposition in HPSG, Manuscript.

King, P. J.: 1994, An expanded logical formalism for Head-Driven Phrase Structure Grammar, *Arbeitspapiere des SFB 340 59*, Universität Tübingen.

Nerbonne, J.: 1992, Constraint-based semantics, *in* P. Dekker and M. Stokhof (eds), *Proceedings of the 8th Amsterdam Colloquium*, Institute for Logic, Language and Information, pp. 425 – 444.

Pinkal, M.: 1995, Radical underspecification, *Proceedings of the 10th Amsterdam Colloquium*, pp. 587 – 606.

Pollard, C. and Sag, I.: 1987, *Information Based Syntax and Semantics. Vol.1: Fundamentals*, CSLI Lecture Notes 13.

Pollard, C. and Sag, I.: 1994, *Head-Driven Phrase Structure Grammar*, University of Chicago Press.

Reyle, U.: 1993, Dealing with ambiguities by underspecification: Construction, representation and deduction, *Journal of Semantics* **10**(2), 123 – 179.

Richter, F. and Sailer, M.: 1996, Syntax für eine unterspezifizierte Semantik: PP-Anbindung in einem deutschen HPSG-Fragment, *in* S. Mehl, A. Mertens and M. Schulz (eds), *Präpositionalsemantik und PP-Anbindung*, Gerhard-Mercator-Universität Duisburg, pp. 39 – 47.