BMBF

Verbmobil

Verbundvorhaben

Predicate Logic Unplugged

Johan Bos

Universität des Saarlandes

Vm

Report 103

Februar 1996

Februar 1996

Johan Bos

Computerlinguistik
Bau 17.2
Universität des Saarlandes
66041 Saarbrücken

Tel.: (0681) 302 - 4680
Fax: (0681) 302 - 4351
e-mail: {bos}@coli.uni-sb.de

Gehört zum Antragsabschnitt: 8.1

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Förderkennzeichen 01 IV 101 R gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei dem Autor.

Contents

1 Introduction

In this paper we describe the syntax and semantics of a description language for underspecified semantic representations. This concept is discussed in general and in particular applied to Predicate Logic and Discourse Representation Theory.

The reason for exploring *underspecified representations* as suitable semantic representations for natural language expressions emerges directly from practical natural language processing applications. The so-called *Combinatorial Explosion Puzzle*, a well known problem in this area, can succesfully be tackled by using underspecified representations. The source of this problem, scopal ambiguities in natural language expressions, is discussed in section 2.

The core of the paper presents *Hole Semantics*. This is a general proposal for a framework, in principle suitable for any logic, where underspecified representations play a central role. There is a clear separation between the object language (the logical language one is interested in) and the meta language (the language that describes and interprets underspecified structures). It has been noted by various authors that the meaning of an underspecified semantic representation cannot be expressed in terms of a disjunction of denotations, but rather as a *set* of denotations (cf. Poesio 1994). We support this view, and use it as underlying principle for the definition of the semantic interpretation function of underspecified structures. Section 3 is an informal introduction to Hole Semantics, and in section 4 things are formally defined. In section 5 we apply Hole Semantics to Predicate Logic, resulting in an "unplugged" version of (static and dynamic) Predicate Logic. In section 6 we show that this idea easily carries over to Discourse Representation Structures.

A lot of attention has been paid to "underspecified semantics" recently. Strongly related to the work presented here are Quasi Logical Forms (Alshawi 1992, Alsahwi and Crouch 1992), Underspecified Discourse Representation Structures (UDRSs) (Reyle 1993), Minimal Recursion Semantics (Egg and Lebeth 1995), and further (Poesio 1994, Muskens 1995, Pinkal 1995). The work presented here provides a straightforward syntax and semantics for a *general* kind of scopally underspecified representations.

2 Natural Language Ambiguities

In every day life, people communicate with each other by uttering true statements, or to put this more generally: they say things that make sense. In a situation where a speaker utters an utterance p, the hearer tries to interpret p in such a way that p denotes truth (rather than falsity). This probably strongly affects the way ambiguous utterances are processed by human beings. Imagine a situation where someone utters (1):

(1) Do not sleep and pay attention, please!

Utterance (1) is in isolation ambiguous. There is a reading where the negation outscopes “sleep and pay attention” and a reading where negation only has scope over “sleep”. Normally, it is context, intonation, or world knowledge that enables a person to select the appropriate reading. Disambiguation is not the topic of this paper. What we are interested in is what introduces these ambiguities, how we represent ambiguities in a logical representation, and how we interpret these representations.

Ambiguities in natural language are caused by different sources, such as predicative ambiguities or structural syntactic ambiguities, but in this paper we will restrict ourselves to semantic scope ambiguities. Among these we find all natural language expressions that, when translated into some logical form, introduce boolean operators, quantifiers, modals, questions, and many more. We will refer to these as operators. When at least two operators appear related to each other in a natural language expression, there is a chance that the expression is ambiguous. In (1) it is the scope of negation (“not”) and conjunction (“and”) that cause the ambiguity.

In the following examples the ambiguity is caused by the scope of implication and conjunction (2), and the scope of the intensional verb and disjunction (3). The absence of prosodic information in these examples make them ambiguous.

(2) If a man walks then he whistles and a woman is happy

(3) Do you want tea or coffee?

Standard examples in the literature on quantifier scope ambiguities and underspecification are (4) and (5). These kinds of examples are traditionally used to provide evidence that human beings do not disambiguate while processing natural language input. While (4) is said to have thousands of readings, it seems very unlikely that humans generate and test every one of them.

(4) A politician can fool most voters on most issues most of the time, but no politician can fool all voters on every single issue all of the time

(5) Everybody is not here

In the previously mentioned references to underspecified semantics, most authors seem to agree on an approach where an underspecified representation plays a central role. Scope ambiguities are not resolved but are put together in a very compact representation.

Of interest is a kind of representation that describes the (complete and sound) logical translations of ambiguous expressions. In this paper we define a semantic representation that is able to express underspecification for any kind of object language. First we sketch the basic idea of underspecified representations, then we move on to precisely defining its ingredients and properties.

3 Underspecified Representations

Semantic representations of natural language expressions are traditionally constructed on the basis of their syntactic analysis. Since expressions can be semantically ambiguous, this is a one-to-many mapping. The idea of underspecified representations is to make this mapping functional, i.e., a one-to-one mapping from syntactic to semantic structure. The interpretation of an underspecified semantic representation is (hence) the set of interpretations that are expressed in it.

The way we define underspecified representations is as follows. Take an object language (the logic in which you are interested), and define the syntax of its basic formulas. We label these formulas for an obvious reason: it will be very easy to talk about them on a meta level. Labels are used as constants. Then we introduce variables over labels, which we will refer to as *holes*, as arguments of scope bearing operators. The last step is to add a set of constraints on the labels and holes, that tell how the different pieces of structures fit together, in such a way that all readings are covered. So what we end up with is a set of labeled formulas, a set of meta variables (holes), and a set of constraints. This is our underspecified representation (UR).

Constraints state relations between the different formulas in UR with respect to scope. For example, it is possible to say that a formula (with label l) is in the scope of an operator (with hole h) by $l \leq h$. This constraint forces l to be directly or indirectly in the scope of h of the relevant operator (e.g., l is in the scope of an operator with label l', and l' is in the scope of h).

So, metaphorically speaking, *holes* underspecify scope in an UR. In order to give URs a non-ambiguous interpretation, the holes should be plugged with the (labeled) formulas of UR in such a way that all the constraints of UR are satisfied. We illustrate this idea with a simple example, where we take Propositional Logic as object language. We use the following notational conventions: holes are represented by h_i, i an index. We label a formula ϕ as l_i: ϕ, where l is a label with index i.

Consider again (1). Assume that there is some syntactic analysis for it on which we build our UR. Translate the negation ("do not") as "$l_1 : \neg h_1$", and the disjunction ("or") as "$l_2 : h_2 \vee h_3$". Take "$l_3 : S$" as translation for "sleep", and "$l_4 : P$" for "pay attention". As variable for "widest scope" we take "h_0". Finally, we set our constraints in the following way: "sleep" should be in the scope of negation ($l_3 \leq h_1$) and in the scope of the left disjunct ($l_3 \leq h_2$), "pay attention" in the scope of the right disjunct ($l_4 \leq h_3$), and the operators l_1 and l_2 can both take wide scope ($l_1 \leq h_0$ and $l_2 \leq h_0$). Then a graphical representation of the UR for (1) is (the constraints are graphically realized by arrows):

(6)
$$\begin{array}{cccccccccc}
 & & h_0 & & & & & & & \\
 & \nearrow & & \nwarrow & & & & & & \\
l_1 : \neg h_1 & & & & l_2 : & h_2 & \vee & h_3 & & \\
 & \nwarrow & & \nearrow & & & & & \nwarrow & \\
 & & l_3 : S & & & & & & & l_4 : P
\end{array}$$

Now we will pay attention to the interpretation of (6), by taking into consideration the possible mappings from holes to labels (pluggings). In other words: a plugging is a bijective assignment function, with the set of holes as scope and the set of labels as range. In this example, there are exactly two possible pluggings, P_1 and P_2:

(7) $P_1 : \{h_0 = l_1, h_1 = l_2, h_2 = l_3, h_3 = l_4\}$

(8) $P_2 : \{h_0 = l_2, h_1 = l_3, h_2 = l_1, h_3 = l_4\}$

The reader may check that these are indeed the only admissible pluggings: for hole h_0 label l_3 does not come into question since it will never be in the scope of h_1 or h_2 and hence not all constraints would be satisfied; for hole h_3, the only suitable candidate is l_4. The two pluggings (7) and (8) correspond to the object language formulas in (9) and (10) respectively. The interpretation of (6) is a set, containing the interpretation of (9) and that of (10).

(9) $(\neg(sleep \vee pay\_attention))$

(10) $((\neg sleep) \vee pay\_attention)$

We summarize and discuss this section shortly. An UR consists of a set of labeled formulas, a set of meta variables that represent scope (holes), and a set of constraints on these. The idea of labeling is taken directly from Reyle's Underspecified DRSs (Reyle 1993). A notable difference is that Reyle uses labels for DRSs, whereas we use them for smaller logical units, since this gives us an advantage with respect to the descriptive power of URs. In this section we sketched by way of an example what URs are. The next section formalizes these ideas.

4 Hole Semantics

The underspecified representations proposed in the previous section are now subject to more formal specifications. We define the syntax and semantics of an UR, and also the notions *proper* UR, *consistent* UR, and *possible pluggings*. Let us start with the syntax.

Definition 1: **SYNTAX UR**

Let H a set of variables over formulas, L the set of labeled X-formulas, and C the set of constraints on $H \cup L$. Then an UR $U = \langle H, L, C \rangle$

In the rest of the paper, we will use U for an underspecified representation and H_U, L_U, and C_U to refer to the holes, labeled formulas and constraints of U respectively. The syntax of expressions in L_U obviously depends on the object language, therefore we do not pay any attention to it just now, but postpone it to the next section, where we take predicate logic as our target language. We use P (sometimes indexed) for pluggings, which are total assignments from holes to labels.

Let us first make a brief excursion to lattice theory, from which we borrow some principles. We can view a $U = \langle H, L, C \rangle$ as a join semi-lattice $\langle H_U \cup L_U, \leq \rangle$. $\leq$ is reflexive, transitive and antisymmetric and therefore a partial order. For any $k_i, k_j \in H_U \cup L_U$, there is a k such that k is the least upper bound of k_i and k_j. End of excursion. Now we define subordination for labels or holes in U as SUB(k,k'), meaning "k is subordinated to k'", or k' subordinates k.

Definition 2: **SUBORDINATION (SUB)**

Let l be a label, h a hole, k a hole or a label of U. Then:

1. $\text{SUB}_U(k, k)$;
2. $\text{SUB}_U(k, k')$ if there is a $k \leq k' \in C_U$
3. $\text{SUB}_U(h, l)$ if there is a ϕ such that $l : \phi \in L_U$ and h is an argument of $l : \phi$ and it is not the case that $\text{SUB}_U(l, h)$;
4. $\text{SUB}_U(k, k')$ if there is a k'' such that $\text{SUB}_U(k, k'')$ and $\text{SUB}_U(k'', k')$;
5. SUB is only defined on the basis of 1-4.

The second clause is the explicit way of defining subordination: if there is a constraint $\leq$ present in U. The first clause represents reflexivity, the third clause defines subordination on labeled formulas that have holes as arguments. The fourth clause expresses transitivity. With SUB we can define a *proper* UR.

Definition 3: **PROPER UR**

An UR U is *proper* iff for all $k, k' \in H_U \cup L_U$ it is the case that there is a k'' such that $\text{SUB}_U(k,k'')$ and $\text{SUB}_U(k',k'')$.

A proper UR is one which describes a join semi-lattice. Yet we are able to define what, with respect to a plugging, a *consistent* UR is, using the following notational convention: for any $k \in H_U \cup L_U$, we define $I_P(k) = P(k)$ iff $k \in H_U$, and $I_P(k) = k$ iff $k \in L_U$. A consistent UR is an UR which is proper, taking pluggings into account.

Definition 4: **CONSISTENT UR**

$\text{CONS}_{U,P}$ iff for all k, k' such that $\text{SUB}(k,k')$, it is the case that either $I(k) = I(k')$, or $I(k) \neq I(k')$ and $\text{SUB}(k',k)$ is not supported.[1]

We have not yet defined what possible pluggings are. Pluggings are, as we have discussed in the previous section, bijective functions from holes to labels. A plugging for an UR U is *possible*, if the UR, with respect to this plugging, is consistent. In other words, when the underspecified representation, taking the plugging into account, has the properties of a join semi-lattice. Since we have already defined what a consistent UR is, defining possible pluggings is an easy job.

Definition 5: **POSSIBLE PLUGGING (PP)**

$\text{PP}_U = \{P \mid \text{CONS}_{P,U}\}$

A plugging is possible, if U is consistent with respect to this plugging. We will illustrate this with two examples. First example: suppose that $U = < \{h_0\}, \{l_1 : \phi\}, \{l_1 \leq h_0\}$ for some formula ϕ. Hence, $\text{SUB}(l_1,l_1)$, $\text{SUB}(h_1,h_1)$, and $\text{SUB}(l_1,h_1)$ are valid. Then a possible plugging P for U is one such that $\text{P}(h_0)=l_1$, since $\text{CONS}_{U,P}$ holds.

Second example: consider the following constraints of an UR: $\{h_1 \leq l_1, h_2 \leq l_2, l_2 \leq h_0, l_1 \leq h_0, l_3 \leq h_1, l_3 \leq h_2\}$, then a plugging P where $P(h_0)=l_3$, $P(h_1)=l_2$, and $P(h_2)=l_1$ is not possible. The UR to which these constraints belong is not consistent, since, for example, $\text{SUB}(l_3,h_2)$ and $\text{SUB}(h_2,h_0)$ are valid and with P lead to "$\text{SUB}(l_3,l_1)$" and "$\text{SUB}(l_1,l_3)$", violating antisymmetry.

So far, so good. We have defined the syntax, properness, and consistency of an UR. For the semantic interpretation of an UR we need to be able to address the label or hole that subordinates all others. We call this TOP, and define it as follows.

[1] Read this giving disjunction scope over conjunction!

Definition 6: **TOP**

$\text{TOP}_{P,U} = I(k)$ iff $k \in H_U \cup L_U$ and there is no k' such that $k' \in H_U \cup L_U$, $k \neq k'$, and $\text{CONS-SUB}_{P,C_U}(k, k')$.

The semantic interpretation of an UR is that of its TOP. As interpretation function for URs, with respect to a model **M** we will use $[\![.]\!]_{\mathbf{M}}$, as to avoid confusion with the interpretation function of the object language, for which we will adopt the traditional $[\![\cdot]\!]_{\mathbf{M},P}$.

Definition 7: **SEMANTICS UR** ($[\![.]\!]$)

$$[\![U]\!]_{\mathbf{M}} = \{ \ [\![\text{TOP}_{P,U}]\!]_{\mathbf{M},P} \mid P \in \text{PP}_U \ \}$$

This definition states that the interpretation of an underspecified representation U^X, is the set of object language denotations, as many as there are possible pluggings for U^X. For some reasons it might be an advantage to redefine this function. For example, when the object denotations are truth values, the interpretation of an UR for this particular object language has three different values: $\{0\}$, $\{1\}$, and $\{0,1\}$. This approach is too weak to capture the fact that an UR might have more than one interpretation with the same denotation. This situation can be avoided if we relate the object denotation to a plugging, as we do in our revised semantic interpretation function.

Definition 8: **SEMANTICS UR (revised)** ($[\![.]\!]^*$)

$$[\![U]\!]^*_{\mathbf{M}} = \{ < P, \ [\![\text{TOP}_{P,U}]\!]_{\mathbf{M},P} > \mid P \in \text{PP}_U \ \}$$

Here $[\![.]\!]^*$ is defined as a function from URs to a set of pairs of pluggings and object language denotations. And this ends the general specification of Hole Semantics. In the next section we will apply Hole Semantics to Predicate Logic.

5 Predicate Logic "Unplugged"

In this section we take Predicate Logic as object language, resulting in Predicate Logic Unplugged (PLU). Given the framework of Hole Semantics described in the previous section, we only need to define the syntax of PLU formulas and their model interpretation. Taking as convention that terms (written as $t_1, ..., t_n$) are either object language variables or constants, PLU formulas are defined as follows:

Definition 9: **Syntax PLU formulas**

1. If h_i, h_j are holes, then $h_i \rightarrow h_j$, $h_i \vee h_j$, $h_i \wedge h_j$ are PLU formulas;
2. If h is a hole, then $\neg h$ is a PLU formula;
3. If x is an object language variable, h a hole, then $\forall x\ h$ and $\exists x\ h$ are PLU formulas;
4. If R is a predicate symbol for an n-place predicate, and $t_1, ..., t_n$ are terms, then $Rt_1, ..., t_n$ is a PLU formula;
5. Nothing else is a PLU formula.

The syntax of PLU formulas is in principle the same as that of ordinary Predicate Logic, with the exception that holes in places, where normally PL-formulas are found, are introduced. We will illustrate Predicate Logic Unplugged with an example. Consider (5), repeated here for convenience as (11), and its translation (12) in U^{PL}.

(11) Everybody is not here

$$(12) \quad < \left\{ \begin{array}{l} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \right\}, \left\{ \begin{array}{l} l_1 : \forall x\ h_1 \\ l_2 : h_2 \rightarrow h_3 \\ l_3 : Bx \\ l_4 : \neg h_4 \\ l_5 : Hx \end{array} \right\}, \left\{ \begin{array}{l} l_1 \leq h_0 \\ l_2 \leq h_1 \\ h_1 \leq h_2 \\ l_3 \leq h_2 \\ h_2 \leq l_3 \\ l_4 \leq h_0 \\ l_5 \leq h_3 \\ l_5 \leq h_4 \end{array} \right\} >$$

Note that $I(h_0)$ is TOP, and the label which is plugged into this h_0 will receive widest scope. The other holes in (12) are introduced by the scope bearing operators (universal quantifier, negation, and implication). Further, notice that we constrain l_3 to be directly in the scope of h_2 via the constraints $l_3 \leq h_2$ and $h_2 \leq l_3$, and this is also the case for l_2 and h_1. These extra constraints exclude unwanted readings. In a graphical representation, the UR looks like:

$$(13) \quad \begin{array}{ccccc} & & h_0 & & \\ & \nearrow & & \nwarrow & \\ l_1 : \forall x h_1 & & & & \\ \downarrow\uparrow & & & & l_4 : \quad \neg h_4 \\ l_2 : h_2 \rightarrow h_3 & & & & \\ \downarrow\uparrow & \nwarrow & & \nearrow & \\ l_3 : Bx & & l_5 : Hx & & \end{array}$$

There are two pluggings (the interested reader may verify this). Plugging (14) interprets (12) as giving the universal quantifier wide scope, outscoping negation. The corresponding formula in predicate logic is (15), which is true in a model where all persons (in the relevant domain, of course) do not have the property being at the speaker's location.[2]

(14) $P_1 : \{h_0 = l_1, h_1 = l_2, h_2 = l_3, h_3 = l_4, h_4 = l_5\}$

(15) $\forall x(Bx \rightarrow \neg Hx)$

Plugging (16) interprets (12) as negation outscoping the universal quantifier. In a model where there is some person that is not at the speaker's location this interpretation denotes truth. A corresponding formula in predicate logic is (17).

(16) $P_2 : \{h_0 = l_4, h_1 = l_2, h_2 = l_3, h_3 = l_5, h_4 = l_1\}$

(17) $\neg\forall x(Bx \rightarrow Hx)$

The model interpretation of PLU can be sketched as follows. Call $[\![.]\!]^{PLU}_{M,P}$ the interpretation function for PLU formulas, and $\mathbf{M} = < D, F >$ a model. D is the domain (a nonempty set) and F an interpretation function ($F(d) \in D$ if d is a constant, and $F(R) \subset D^n$ for an n-place predicate symbol R). As usual, we use g and g' for total assignment functions. For a term t, $[\![t]\!]_{M,g}$ is $g(t)$ if t is a variable, and $F(t)$ is t is a constant.

Definition 10: **Interpretation Function for PLU** ($[\![.]\!]^{PLU}$)

1. $[\![h_i \rightarrow h_j]\!]^{PLU}_{\mathbf{M},P,g} = 1$
   iff $[\![h_i]\!]^{PLU}_{\mathbf{M},P,g} = 0$ or $[\![h_j]\!]^{PLU}_{\mathbf{M},P,g} = 1$

2. $[\![h_i \vee h_j]\!]^{PLU}_{\mathbf{M},P,g} = 1$
   iff $[\![h_i]\!]^{PLU}_{\mathbf{M},P,g} = 1$ or $[\![h_j]\!]^{PLU}_{\mathbf{M},P,g} = 1$

3. $[\![h_i \wedge h_j]\!]^{PLU}_{\mathbf{M},P,g} = 1$
   iff $[\![h_i]\!]^{PLU}_{\mathbf{M},P,g} = [\![h_j]\!]^{PLU}_{\mathbf{M},P,g} = 1$

[2] We translate the deictic property "being here" simply as H, for convenience.

4. $[\![\neg h_i]\!]^{PLU}_{\mathbf{M},P,g} = 1$
   iff $[\![h_i]\!]^{PLU}_{\mathbf{M},P,g} = 0$

5. $[\![\forall x\ h_i]\!]^{PLU}_{\mathbf{M},P,g} = 1$
   iff for all $d \in D$ it is the case that $[\![h_i]\!]^{PLU}_{\mathbf{M},P,g[d/x]} = 1$

6. $[\![\exists x\ h_i]\!]^{PLU}_{\mathbf{M},P,g} = 1$
   iff for at least some $d \in D$ it is the case that $[\![h_i]\!]^{PLU}_{\mathbf{M},P,g[d/x]} = 1$

7. $[\![Rt_1, ..., t_n]\!]^{PLU}_{\mathbf{M},P,g} = 1$
   iff $< [\![t_1]\!]_{M,g}, ..., [\![t_n]\!]_{M,g} > \in F(R)$

Using definition 11, we are able to define an interpretation of underspecified representations of PLU itself, since this definition does not depend on the object language. In fact, using the syntax of PLU, it is fairly easy to define a *dynamic* underspecified Predicate Logic.

Suppose that $[\![.]\!]^{DPLU}_{\mathbf{M},P}$ is the interpretation function that interprets labeled PLU-formulas as in Dynamic Predicate Logic (DPL) (Groenendijk and Stokhof 1991). An assignment g is a function from variables to elements of D, $g'[x]g$ means that g' is an x-variant of g.

Definition 11: **Interpretation Function for DPLU** ($[\![.]\!]^{DPLU}$)

1. $[\![h_i \rightarrow h_j]\!]^{DPLU}_{\mathbf{M},P} =$
   $\{\langle g, g\rangle \mid \forall g'(\langle g, g'\rangle \in [\![h_i]\!]^{DPLU}_{\mathbf{M},P} \rightarrow \exists g''(\langle g', g''\rangle \in [\![h_j]\!]^{DPLU}_{\mathbf{M},P}))\}$

2. $[\![h_\vee h_j]\!]^{DPLU}_{\mathbf{M},P} =$
   $\{\langle g, g\rangle \mid \exists g'(\langle g, g'\rangle \in [\![h_i]\!]^{DPLU}_{\mathbf{M},P} \vee \langle g, g'\rangle \in [\![h_j]\!]^{DPLU}_{\mathbf{M},P})\}$

3. $[\![h_i \wedge h_j]\!]^{DPLU}_{\mathbf{M},P} =$
   $\{\langle g, g'\rangle \mid \exists g''(\langle g, g''\rangle \in [\![h_i]\!]^{DPLU}_{\mathbf{M},P} \& \langle g'', g'\rangle \in [\![h_j]\!]^{DPLU}_{\mathbf{M},P})\}$

4. $[\![\neg h_i]\!]^{DPLU}_{\mathbf{M},P} =$
   $\{\langle g, g\rangle \mid \neg\exists g'(\langle g, g'\rangle \in [\![h_i]\!]^{DPLU}_{\mathbf{M},P})\}$

5. $[\![\forall x\ h_i]\!]^{DPLU}_{\mathbf{M},P} =$
   $\{\langle g, g\rangle \mid \forall g'(g'[x]g \rightarrow \exists g''(\langle g', g''\rangle \in [\![h_i]\!]^{DPLU}_{\mathbf{M},P}))\}$

6. $[\![\exists x\ h_i]\!]^{DPLU}_{\mathbf{M},P} =$
   $\{\langle g, g'\rangle \mid \exists g''(g''[x]g \& \langle g'', g'\rangle \in [\![h_i]\!]^{DPLU}_{\mathbf{M},P})\}$

7. $[\![Rt_1, ..., t_n]\!]^{DPLU}_{\mathbf{M},P} =$
   $\{\langle g, g\rangle \mid < [\![t_1]\!]_{M,g}, ..., [\![t_n]\!]_{M,g} > \in F(R)\}$

6 Underspecified Discourse Representation Structures

As stressed before, Hole Semantics is in principle independent of the object language. Besides Predicate Logic, we could also take Discourse Representation Structures (DRSs, as proposed in Discourse Representation Theory (DRT) Kamp and Reyle 1993) as object language, resulting in DRT Unplugged (DRTU). We first define DRTU formulas:

Definition 12: **Syntax DRTU formulas**

1. If h_i, h_j are holes, $k_1, ..., k_n$ holes or labels, then $[\ |h_i \rightarrow h_j]$, $\otimes\{k_1, ..., k_n\}$, $[\ |\neg h_i]$, $[\ |h_i \vee h_j]\}$ are DRTU formulas;
2. If x is a discourse marker, P a symbol for an n-place predicate, then $[x|\ ]$ and $[\ |P(x_1, ..., x_n)]$ are DRTU formulas.
3. Nothing else is a DRTU formula.

Here a DRS is represented as $[D|C]$, D the set of discourse markers, C the set of conditions. The merger ($\otimes$) makes one DRS out of several by taking the union of the domains and the conditions respectively of its argument, a set of DRSs. The definition of $[\![.]\!]^{DRTU}$ can for example be realised along the lines presented in Kohlhase et al. 1995 or Muskens 1993. We will not present it here, but instead give an example. Consider again (2), repeated here as (18). The UR translation is shown in (19).

(18) If a man walks then he whistles and a woman is happy

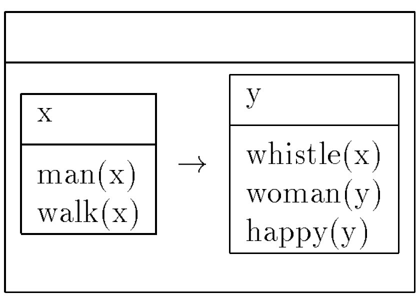
$$(19) \quad < \left\{ \begin{array}{l} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{array} \right\}, \left\{ \begin{array}{l} l_1 : [\ |\ h_1 \rightarrow h_2\ ] \\ l_2 : \otimes\{h_3, h_4\} \\ l_3 : \otimes\{l_4, l_5, l_6\} \\ l_4 : [\ x\ |\ ] \\ l_5 : [\ |\ man(x)\ ] \\ l_6 : [\ |\ walk(x)\ ] \\ l_7 : [\ |\ whistle(x)\ ] \\ l_8 : \otimes\{l_9, l_{10}, l_{11}\} \\ l_9 : [\ y\ |\ ] \\ l_{10} : [\ |\ woman(y)\ ] \\ l_{11} : [\ |\ happy(y)\ ] \end{array} \right\}, \left\{ \begin{array}{l} l_3 \leq h_1 \\ l_7 \leq h_2 \\ l_7 \leq h_3 \\ l_8 \leq h_4 \\ l_1 \leq h_0 \\ l_2 \leq h_0 \end{array} \right\} >$$

There are two possible pluggings for (19), and therefore two readings for (2) available. The first reading (paraphrased as a linear DRS in 21, and for convenience in the more familiar boxed notation 25) triggered by plugging (20) corresponds to the "wide scope disjunction" reading.

$$(20) \quad P_1 : \{h_0 = l_1, h_1 = l_3, h_2 = l_2, h_3 = l_7, h_4 = l_8\}$$

$$(21) \quad [\ |\ [\ x\ |\ man(x)\ walk(x)\ ] \rightarrow [\ y\ |\ whistle(x)\ woman(y)\ happy(y)\ ]\ ]$$

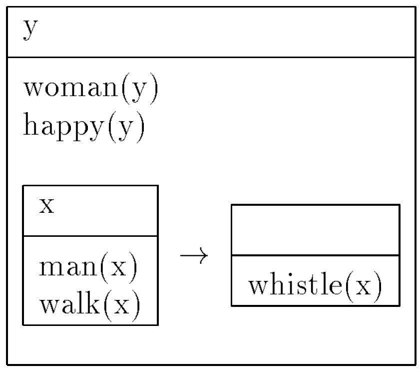
(22)



The other possible plugging (23) results in a reading where conjunction outscopes disjunction. The DRS for this reading is shown in (24) and (25).

$$(23) \quad P_1 : \{h_0 = l_2, h_1 = l_3, h_2 = l_7, h_3 = l_1, h_4 = l_8\}$$

$$(24) \quad [\ y\ |\ woman(y)\ happy(y)\ [\ x\ |\ man(x)\ walk(x)\ ] \rightarrow [\ |\ whistle(x)\ ]\ ]$$

(25) 

7 Conclusion

We proposed a framework for underspecified semantics representations in general, called *Hole Semantics*, and claimed that, due to a clear seperation of object and meta level, it is independent of the object languages. Underspecified Representations in Hole Semantics correspond to a partial descriptions of the semantics, using meta variables (holes) and subordination constraints. We have shown that Hole Semantics can be applied both to Predicate Logic (static and dynamic) and Discourse Representation Theory, with respect to semantically ambiguous scope.

Acknowledgements

This work was partly funded by the German Ministry for Research and Technology (BMBF) under contract 01 IV 101 R. I would like to thank Patrick Blackburn, Emiel Krahmer, David Milward, John Nerbonne, Wojciech Skut, Manfred Pinkal, and my colleagues in the VERBMOBIL project: Björn Gambäck, Scott McGlashan (especially since he provided the title of this paper), Christian Lieske, Yoshiki Mori, CJ Rupp, Michael Schiehlen, and Karsten Worm for helpful comments and discussion. The author is solely responsible for the view expressed in this paper.

References

Alsahwi, H. and Crouch, R.: 1992, Monotonic Semantic Interpretation, in *Proceedings of the 30th Annual Meeting of the ACL*, pp 32–39

Alshawi, H. (ed.): 1992, *The Core Language Engine*, The MIT Press, Cambridge, Massachusetts

Egg, M. and Lebeth, K.: 1995, Semantic underspecification and modifier attachment ambiguities, in J. Kilbury and R. Wiese (eds.), *Integrative Ansätze in der Computerlinguistik*, pp 19–24, Düsseldorf, Seminar für Allgemeine Sprachwissenschaft

Groenendijk, J. and Stokhof, M.: 1991, Dynamic Predicate Logic, *Linguistics and Philosophy* 14, 39–100

Kamp, H. and Reyle, U.: 1993, *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*, Kluwer, Dordrecht

Kohlhase, M., Kuschert, S., and Pinkal, M.: 1995, A Type-Theoretic Semantics for λ-DRT, in *Proceedings of the Tenth Amsterdam Colloquium*

Muskens, R.: 1993, A Compositional Discourse Representation Theory, in *Proceedings of the Ninth Amsterdam Colloquium*

Muskens, R. A.: 1995, Order-independence and Underspecification, in J. Groenendijk (ed.), *Ellipsis, Underspecification, Events and More in Dynamic Semantics*, pp 17–34, Dyana Deliverable R2.2.C

Pinkal, M.: 1995, Radical Underspecification, in *Proceedings of the Tenth Amsterdam Colloquium*

Poesio, M.: 1994, Ambiguity, underspecification and discourse interpretation, in H. Bunt, R. Muskens, and G. Rentier (eds.), *International Workshop on Computational Semantics*, University of Tilburg

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