

Minimal Recursion Semantics: An Introduction

ANN COPESTAKE¹, DAN FLICKINGER², CARL POLLARD³ and
IVAN A. SAG⁴

¹Computer Laboratory, University of Cambridge, 15 JJ Thomson Avenue, Cambridge, CB3 0FD, UK (E-mail: aac@cl.cam.ac.uk); ²CSLI, Stanford University, Ventura Hall, Stanford, CA 94305 USA, and Linguistics Institute, University of Oslo, Norway (E-mail: danf@csli.stanford.edu); ³Linguistics Department, Ohio State University, 25 Oxley Hall, 1712 Neil Ave. Mall, Columbus, OH 43210, USA (E-mail: pollard.2@osu.edu); ⁴Department of Linguistics, Stanford University, Stanford, CA 94305, USA (E-mail: sag@csli.stanford.edu)

Abstract. Minimal recursion semantics (MRS) is a framework for computational semantics that is suitable for parsing and generation and that can be implemented in typed feature structure formalisms. We discuss why, in general, a semantic representation with minimal structure is desirable and illustrate how a descriptively adequate representation with a nonrecursive structure may be achieved. MRS enables a simple formulation of the grammatical constraints on lexical and phrasal semantics, including the principles of semantic composition. We have integrated MRS with a broad-coverage HPSG grammar.

Key words: computational semantics, flat semantics, semantic composition, grammar implementation

1. Introduction

Our aim in this paper is to describe an approach to semantic representation for large-scale linguistically-motivated computational grammars of natural language. We believe such grammars should support both parsing and generation, and should be useful for multiple applications, including natural language interfaces of various sorts and machine translation. Our main general criteria for computational semantics are:

Expressive Adequacy. The framework must allow linguistic meanings to be expressed correctly.

Grammatical Compatibility. Semantic representations must be linked cleanly to other kinds of grammatical information (most notably syntax).

Computational Tractability. It must be possible to process meanings and to check semantic equivalence efficiently and to express relationships between semantic representations straightforwardly.

Underspecifiability. Semantic representations should allow underspecification (leaving semantic distinctions unresolved), in such a way as to allow flexible, monotonic resolution of such partial semantic representations.

The first and second criteria are the object of much ongoing work in semantics, only a small subset of which claims to be computational in its aims. But computational linguists have to balance these requirements with those imposed by the third and fourth criteria. Expressive accuracy in particular has often been sacrificed in order to achieve computational tractability. This has been especially noticeable in natural language generation and machine translation, both of which require some straightforward way of decomposing semantic representations and relating partial structures. Conventional forms of standard semantic representation languages have proved problematic for these tasks. As we will discuss below, one major difficulty is in ensuring that a grammar can generate from valid logical forms while keeping the module that constructs these forms independent of the grammar. Another problem is to relate semantic representations, as is required, for instance, in a semantic transfer approach to Machine Translation (MT). Issues of this sort have led some researchers to abandon semantically-driven generation in favor of purely lexically-driven approaches, such as Shake-and-Bake (e.g., Whitelock, 1992). Others have used semantic formalisms which achieve tractability at the cost of expressive adequacy, in particular by omitting the scope of quantifiers and disallowing higher-order predicates in general (e.g., Phillips, 1993; Trujillo, 1995).

Conventional semantic representations are also problematic for parsing, since they require information that cannot be obtained by compositional processing of sentences in isolation. In particular, they require that quantifier scope be specified. However, not only is it very difficult to resolve quantifier scope ambiguities, it is also often unnecessary in applications such as MT, since the resolution of quantifier scope usually has no effect on the translation. Thus there has been considerable work in the last fifteen years on developing representations which allow scope to be underspecified. See, for instance, Alshawi and Crouch (1992), Reyle (1993), Bos (1995), Pinkal (1996) and the papers collected in van Deemter and Peters (1996) (though we will not assume familiarity with any of this work in this paper).

In our view, these issues are linked: a framework for semantic representation which makes it easy to decompose, relate and compare semantic structures also naturally lends itself to scope underspecification. In both cases, the key point is that it should be possible to ignore scope when it is unimportant for the task, at the same time as ensuring that it can be (re)constructed when it is required. In this paper, we describe a framework for representing computational semantics that we have called Minimal Recursion Semantics (MRS).¹ MRS is not, in itself, a semantic theory, but can be most simply thought of as a meta-level language for describing

semantic structures in some underlying object language. For the purposes of this paper, we will take the object language to be predicate calculus with generalized quantifiers.²

The underlying assumption behind MRS is that the primary units of interest for computational semantics are *elementary predications* or EPs, where by EP we mean a single relation with its associated arguments (for instance, *beyond*(x, y)). In general, an EP will correspond to a single lexeme. MRS is a syntactically ‘flat’ representation, since the EPs are never embedded within one another. However, unlike earlier approaches to flat semantics, MRS includes a treatment of scope which is straightforwardly related to conventional logical representations. Moreover, the syntax of MRS was designed so that it could be naturally expressed in terms of feature structures and, as we shall illustrate, smoothly integrated with a feature-based grammar, e.g., one written within the framework of Head-driven Phrase Structure Grammar (HPSG). The point of MRS is not that it contains any particular new insight into semantic representation, but rather that it integrates a range of techniques in a way that has proved to be very suitable for large, general-purpose grammars for use in parsing, generation and semantic transfer (e.g., Copestake et al., 1995; Copestake 1995; Carroll et al., 1999). It also appears that MRS has utility from a non-computational perspective (see Riehemann, 2001; Bouma et al., 1998; and Warner, 2000, among others).

In the next section of the paper, we motivate flat semantics in more detail. In §3 we introduce MRS itself, and in §4 we provide more formal details, and discuss semantic composition. We then turn to the implementation of MRS within a typed feature structure logic. We introduce this in §5 and follow in §6 with concrete examples of the use of MRS within a large HPSG for English. Note that, for ease of exposition of the fundamentals of MRS, we adopt a very simple account of the semantics of many constructions in the first parts of this paper, but §6 describes a fuller approach. §7 discusses related work and in §8 we conclude by recapitulating the main features of MRS and discussing current and future work.

2. Why Flat Semantics?

To illustrate some of the problems involved in using conventional semantic representations in computational systems, we will consider examples which arise in semantic transfer approaches to MT. The term semantic transfer refers to an approach where a source utterance is parsed to give a semantic representation and a transfer component converts this into a target semantic representation. This is then used as input to a generator to produce a string in the target language. Semantic transfer provides a particularly stringent test of a computational semantic representation, especially in a broad-coverage

system. There has been considerable discussion of the problems within the MT literature (Trujillo, 1995) has an especially detailed account), so we will provide only a brief overview here.

To begin with, consider the following trivial example. An English grammar might naturally produce the logical form (LF) in (1a) from *fierce black cat*, while a straightforward transfer from the natural Spanish representation of *gato negro y feroz* shown in (1b) would produce the LF in (1c), which the English grammar probably would not accept:

- (1) a. $\lambda x[\text{fierce}(x) \wedge (\text{black}(x) \wedge \text{cat}(x))]$
 b. $\lambda x[\text{gato}(x) \wedge (\text{negro}(x) \wedge \text{feroz}(x))]$
 c. $\lambda x[\text{cat}(x) \wedge (\text{black}(x) \wedge \text{fierce}(x))]$

The strictly binary nature of $\wedge$ leads to a spurious ambiguity in representation, because the bracketing is irrelevant to the truth conditions. But a generator could only determine this by examining the logical properties of $\wedge$. The problem is that the form of these semantic representations implicitly includes information about the syntactic structure of the phrase, even though this is irrelevant to their semantic interpretation.

In the example above, the individual relations are in a one-to-one equivalence between the languages. But often this is not the case, and then a similar representation problem also affects the transfer component. Consider the possible translation equivalence between (one sense of) the German *Schimmel* and *white horse*, which might be represented as (2a). There is a potential problem in applying the rule to *white English horse* because of the bracketing (see (2b)).

- (2) a. $[\text{white}(x) \wedge \text{horse}(x)] \leftrightarrow \text{Schimmel}(x)$
 b. $[\text{white}(x) \wedge (\text{English}(x) \wedge \text{horse}(x))]$

There are also more complex cases. For instance, the English phrase *beginning of spring* can be translated into German as *Frühlingsanfang*. In (3) we show two possible scopes for the sentence *the beginning of spring arrived* which arise on the assumption that *spring* is taken as contributing a definite quantifier:³

- (3) The beginning of spring arrived.
 $\text{def_q}(x, \text{spring}(x), \text{the}(y, \text{beginning}(y, x), \text{arrive}(y)))$
 $\text{the}(y, \text{def_q}(x, \text{spring}(x), \text{beginning}(y, x)), \text{arrive}(y))$

Even without going into details of the representation in German or of the transfer rules, it should be apparent that it is potentially difficult to make the mapping. If both scopes are allowed by the English grammar, then it is not possible to write a single transfer rule to map from English to German without a powerful transfer language which allows higher-order unification or the equivalent. But if the grammar only allows one scope, the transfer

component needs to know which is the ‘valid’ scope when going from German to English.

We refer the reader who wants further discussion of these issues to Trujillo (1995). Our point here is just that there are two related problems: How do we make it as easy as possible to write general transfer rules, and how do we ensure that the system can generate from the input LF? The second problem has received more attention: it might seem that the generator could try logically equivalent variants of the LF until it finds one that works, but this is not feasible since the logical form equivalence problem is undecidable, even for first order predicate calculus (see Shieber, 1993 for more details). Thus we have the problem of guaranteeing that a grammar can generate from a given logical form. One line which was investigated was the development of isomorphic grammars for the target and source languages (e.g., Landsbergen, 1987). But this has proven to be extremely difficult, if not impossible, to accomplish in a broad-coverage MT system: grammar development is quite complex enough without trying to keep the rules parallel across multiple languages.

The difficulties with isomorphic grammars led some researchers to develop the ‘Shake-and-Bake’ approach to semantic transfer (e.g., Whitelock, 1992). Shake-and-Bake operates by considering transfer between bags of instantiated lexical signs rather than LFs. This avoids problems of the sort we have been considering, which essentially arise from the structure of the LF. However, from our perspective, this is not a good solution, because it is quite specific to MT and it imposes stringent conditions on the grammar (at least in the efficient form of Shake-and-Bake processing described by Poznanski et al., 1995). An alternative is to modify the form of the semantic representation, in particular to use a non-recursive, or flat representation such as those developed by Phillips (1993) or Trujillo (1995) (the latter uses flat semantic structures in conjunction with Shake-and-Bake processing). In (4) we show structures for some of our previous examples in a flat semantic representation roughly equivalent to that used by Trujillo:

- (4) a. fierce black cat
 fierce(x), black(x), cat(x)
- b. the beginning of spring arrives
 the(y), beginning(y , x), def(x), spring(x), arrive(e , y)

Note that the structure is a list of elementary predications, which can be taken as being conjoined. It should be easy to see why this representation makes the representation of transfer rules simpler and avoids at least some of the problems of ensuring that the input LF is accepted by the generator, on the assumption that the generator simply has to accept the members of the flat list in an arbitrary order. Chart generation (e.g., Kay, 1996, Carroll

et al., 1999) is a suitable approach for such a representation since it allows the alternative orderings to be explored relatively efficiently.

But the omission of scope means the representation is semantically inadequate. For example, consider the representation of:

- (5) Every white horse is old.

There is only one possible scope for *every* in this sentence, which is shown in (6) using generalized quantifiers:

- (6) $\text{every}(x, \text{white}(x) \wedge \text{horse}(x), \text{old}(x))$

We should, therefore, be able to retrieve this reading unambiguously from the semantic representation that the grammar constructs for this sentence. However, if we have the totally flat structure shown in (7a), it is impossible to retrieve the correct reading unambiguously, because we would get the same structure for (7b), for instance.

- (7) a. $\text{every}(x), \text{horse}(x), \text{old}(x), \text{white}(x)$
 b. Every old horse is white.

Representations which completely omit scope lose too much information for some applications, even though they are suitable for MT (at least up to a point). They are also clearly inadequate from the perspective of theoretical linguistics. We therefore want an approach which is in the spirit of Trujillo and Phillip's work, but we require a flat representation that preserves sufficient information about scope to be able to construct all and only the possible readings for the sentence. We can also observe that, although full logical equivalence is going to be undecidable for any language that is sufficiently powerful to represent natural language, it seems that this may not be the most desirable notion of equivalence for many problems in computational linguistics, including generation in an MT system. It is apparent from the work on flat semantics and on Shake-and-Bake that the primary interest is in finding the correct lexemes and the relationships between them that are licensed in a very direct way by the syntax. Thus, what we require is a representation language where we can ignore scope when it is irrelevant, but retrieve scopal information when it is needed. Of course this also suggests a language that will support underspecification of quantifier scope during parsing, though we won't motivate the need for this in detail, since it has been amply discussed by various authors (e.g., Hobbs, 1983).

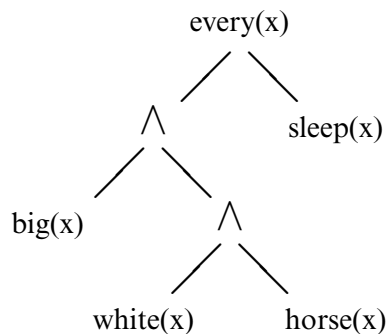
Before we go into detail about how MRS fulfills these requirements, we should emphasize that, although we have concentrated on motivating flat semantics by considering semantic transfer, we believe that the lessons learned are relevant for other applications in NLP. Shieber (1993) discusses logical form equivalence with respect to generation. In small-scale, domain-specific generation applications, the developer can tune the grammar or

the module which produces the semantic representation in order to try and ensure that the generator will accept the LF. But this is somewhat like the isomorphic grammar approach in MT and we believe that, for a large-scale, general-purpose grammar, it is impractical to do this without adopting some form of flat representation.

3. MRS Representation

We will approach our informal description of MRS by starting off from a conventional predicate calculus representation with generalized quantifiers and discussing a series of modifications that have the effect of flattening the representation. It will be useful to regard the conventional representation as a tree, with scopal arguments forming the branches. See, for instance, (8):

- (8) a. Every big white horse sleeps.
 b. $\text{every}(x, \wedge(\text{big}(x), \wedge(\text{white}(x), \text{horse}(x))), \text{sleep}(x))$
 c.



This simple tree notation is straightforwardly equivalent to the conventional notation, assuming that:

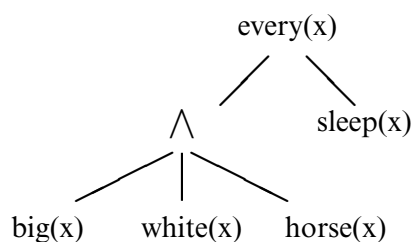
1. All connectives are represented using a prefixed notation and can be treated syntactically like (scopal) predicates. For instance, we can write $\wedge(\text{white}(x), \text{horse}(x))$ rather than $\text{white}(x) \wedge \text{horse}(x)$.
2. All non-scopal arguments in a predication precede all scopal arguments.
3. The position in the scopal argument list is paralleled by the left-to-right ordering of daughters of a node in the tree.⁴

As we mentioned above, we assume a three-argument notation for generalized quantifiers, where the first argument corresponds to the bound variable. We will use the terminology *restriction* and *body* to refer to the two set arguments, where we use *body* rather than *scope* to avoid confusion between this and the more general concept of scope.

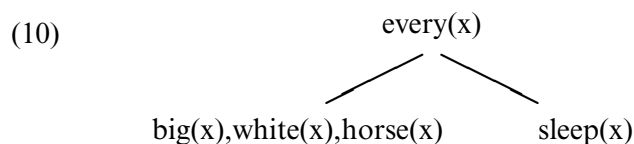
Given that, as discussed in the previous section, binary $\wedge$ gives rise to spurious ambiguities, we can improve on this representation by allowing $\wedge$

to be n-ary. For instance, when representing (8a), rather than (8b)/(8c), we can use (9a)/(9b):

- (9) a. $\text{every}(x, \bigwedge(\text{big}(x), \text{white}(x), \text{horse}(x)), \text{sleep}(x))$
 b.



Our next move is to simplify this notation by omitting the conjunction symbol, on the basis of a convention whereby a group of elementary predicates on a tree node is always assumed to be conjoined. Thus the equivalent of the expression above is shown in (10):



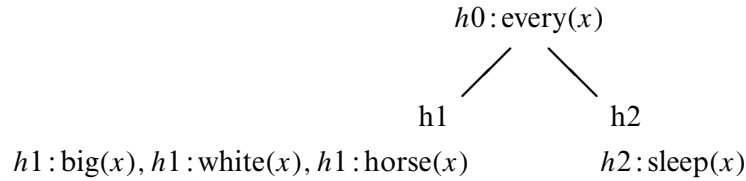
Notice that order of the elements within the group is not semantically significant. However, the structure is a *bag*, rather than a set, because there might be repeated elements.

Besides notational simplicity, the reason for making conjunction implicit is to make it easier to process the sort of examples that we saw in the last section. This special treatment is justifiable because logical conjunction has a special status in the representation of natural language because of its general use in composing semantic expressions. If the other logical connectives ($\vee$, etc.) are used directly at all in semantic representation, they are restricted to occurring in quite specific, lexically-licensed, contexts. Our use of groups of EPS is reminiscent of the use of sets of conditions in Discourse Representation Theory (Kamp and Reyle, 1993).

To take the next step to a flat representation, we want to be able to consider the nodes of a tree independently from any parent or daughters. In MRS, this is done by reifying the links in the tree, using tags which match up scopal argument positions with the EPS (or conjunctions of EPS) that fill them. We refer to these tags as *handles*, since they can be thought of as enabling us to grab hold of an EP. Each EP has a handle which identifies it as belonging to a particular tree node (henceforth *label*), and if it is scopal, it will have handles in its scopal argument positions. We will use h1,

h2 etc to represent the handles, and show labels as preceding an EP with : separating the label and the EP. For instance, the equivalent of (10) can be drawn as an unlinked tree as shown in (11):

(11)



If multiple EPS have the same label, they must be on the same node and therefore conjoined. Since position in the unlinked tree drawing in (11) is conveying no information which isn't conveyed by the handles (other than to specify the order of scopal arguments), we can use a simple flat list of labeled EPS as in (12).

(12) $h0: \text{every}(x, h1, h2), h1: \text{big}(x), h1: \text{white}(x), h1: \text{horse}(x), h2: \text{sleep}(x)$

The use of the same handle to label all EPS in a conjunction means we don't need any extra syntactic devices to delimit the conjuncts in the list of EPS.

Formally we actually have a bag of EPS, since the order is semantically irrelevant, but not a set, since there might be repeated elements.⁵ The structure in (13) is equivalent to the one given in (12).

(13) $h1: \text{white}(x), h0: \text{every}(x, h1, h2), h1: \text{big}(x), h2: \text{sleep}(x), h1: \text{horse}(x)$

The choice of particular names for handles is also semantically irrelevant, so we could equally well write (14) (which is an alphabetic variant of (13)).

(14) $h0: \text{white}(x), h1: \text{every}(x, h0, h3), h0: \text{big}(x), h3: \text{sleep}(x), h0: \text{horse}(x)$

This flat representation facilitates considering the EPS individually. To derive a flat representation something like Trujillo's (modulo the use of event variables, which we will return to in §6), we just ignore the handles. However, the representation shown here so far is just a syntactic variant of the notation we started with, since if we want to draw a conventional tree we just plug the pieces together according to the handles. We can then reconstruct a conventional representation, as before. There is one caveat here: we have left open the issue of whether the relations themselves can have structure. If they cannot, then the MRS notation as described so far does not allow for predicate modifiers. For instance, there is no way of specifying something like *very(big)(x)*. For the purposes of the current discussion, we will ignore this issue and assume that predicate modifiers are not allowed in the object language.

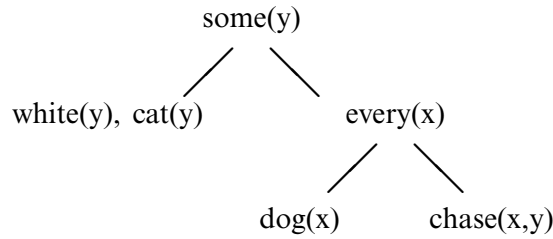
These flat labeled structures are MRSs, but the reason why MRS is not a notational variant of the conventional representation is that it allows underspecification of the links in order to represent multiple scopes (rather than requiring that each handle in an argument position also be the label of some EP). We use the term *handle* for the tags in both fully specified and underspecified links.

To illustrate scope underspecification informally, consider the representation of the sentence in (15):

- (15) Every dog chases some white cat.

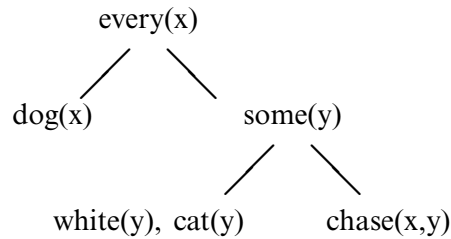
This has the fully specified readings shown in (16) (wide scope *some*) and in (17) (wide scope *every*), where in both cases we show the conventional notation, the tree (with implicit conjunction) and the MRS equivalent:

- (16) a. $\text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(x, y)))$
b.



- c. $h1: \text{every}(x, h3, h4), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y),$
 $h5: \text{some}(y, h7, h1), h4: \text{chase}(x, y)$

- (17) a. $\text{every}(x, \text{dog}(x), \text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{chase}(x, y)))$
b.



- c. $h1: \text{every}(x, h3, h5), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y),$
 $h5: \text{some}(y, h7, h4), h4: \text{chase}(x, y)$

Notice that, in terms of the MRS representation, the only difference is in the handles for the body arguments of the two quantifiers. So if we want

to represent this as a single MRS structure, we can do so by representing the pieces of the tree that are constant between the two readings, and stipulating some constraints on how they may be joined together. The basic conditions on joining the pieces arise from the object language: (1) there must be no cases where an argument is left unsatisfied and (2) an EP may fill at most one argument position. Given these conditions, we can generalize over these two structures, as shown in (18):

- (18) $h1: \text{every}(x, h3, h8), h3: \text{dog}(x), h7: \text{white}(y), h7: \text{cat}(y),$
 $h5: \text{some}(y, h7, h9), h4: \text{chase}(x, y)$

In this example, we have $h8$ and $h9$ as the handles in the body positions of the quantifiers. Following Bos (1995), we will sometimes use the term *hole* to refer to handles which correspond to an argument position in an elementary predication, in contrast to labels.

In terms of the tree representation, what we have done is to replace the fully specified trees with a partially specified structure. Linking the pieces of the representation can be done in just two ways, to give exactly these two trees back again, provided the restrictions on valid predicate calculus formulas are obeyed. That is, we can either link the structures so that $h8 = h5$ and $h9 = h4$, which gives the reading where *every* outscopes *some* (as in (17)), or link them so that $h8 = h4$ and $h9 = h1$, which gives the reading where *some* outscopes *every* (as in (16)). But we couldn't make $h8 = h9 = h4$, for instance, because the result would not be a tree. We will make the linking constraints explicit in the next section. It will turn out that for more complex examples we also need to allow the grammar to specify some explicit partial constraints on linkages, which again we will consider below.

So each analysis of a sentence produced by a grammar corresponds to a single MRS structure, which in turn corresponds to a (non-empty) set of expressions in the predicate calculus object language. If the set contains more than one element, then we say that the MRS is underspecified (with respect to the object language).⁶

So, to summarize this section, there are two basic representational features of MRS. The most important is that we reify scopal relationships as handles so that syntactically the language looks first-order. This makes it easy to ignore scopal relations when necessary, and by allowing the handles to act as variables, we can represent underspecification of scope. The second feature is to recognize the special status of logical conjunction and to adapt the syntax so it is not explicit in the MRS. This is less important in terms of formal semantics and we could have adopted the first feature, but kept an explicit conjunction relation. However implicit conjunction combined with scope reification considerably facilitates generation and transfer using MRS (see Copstake et al., 1995; Carroll et al., 1999).

4. A More Detailed Specification of MRS

Although sophisticated accounts of the semantics of underspecified scope have been proposed, for our current purposes it is adequate to characterize the semantics of MRS indirectly by relating it to the underlying object language. In this section, we will therefore show more precisely how an MRS is related to a set of predicate calculus expressions. In §4.1 we go over some basic definitions that correspond in essential respects to concepts we introduced informally in the last section. In §4.2 we then go on to consider constraints on handles and in §4.3 we introduce an approach to semantic composition.

4.1. BASIC DEFINITIONS

We begin by defining EPS as follows:

DEFINITION 1 (Elementary predication (EP)). An elementary predication contains exactly four components:

1. a handle which is the label of the EP
2. a relation
3. a list of zero or more ordinary variable arguments of the relation
4. a list of zero or more handles corresponding to scopal arguments of the relation (i.e., holes)

This is written as `handle: relation(arg1 ... argn, sc-arg1 ... sc-argm)`. For example, an EP corresponding to *every* is written as `h2: every(y, h3, h4)`.

We also want to define (implicit) conjunction of EPS:

DEFINITION 2 (EP conjunction). An EP conjunction is a bag of EPS that have the same label.

We will say a handle labels an EP conjunction when the handle labels the EPS which are members of the conjunction.

We can say that an EP *E* *immediately outscopes* another EP *E'* within an MRS if the value of one of the scopal arguments of *E* is the label of *E'*. The *outscopes* relation is the transitive closure of the ‘immediately outscopes’ relationship. This gives a partial order on a set of EPS. It is useful to overload the term ‘outscopes’ to also describe a relationship between EP conjunctions by saying that an EP conjunction *C* immediately outscopes another EP conjunction *C'* if *C* contains an EP *E* with an argument that is the label of *C'*. Similarly, we will say that a handle *h* immediately outscopes *h'* if *h* is the label of an EP which has an argument *h'*.

With this out of the way, we can define an MRS structure. The basis of the MRS structure is the bag of EPS, but it is also necessary to include two extra components.

1. The *top handle* of the MRS corresponds to a handle which will label the highest EP conjunction in all scope-resolved MRSs which can be derived from this MRS.
2. *Handle constraints* or *hcons* contains a (possibly empty) bag of constraints on the outscopes partial order. These will be discussed in §4.2, but for now we'll just stipulate that any such constraints must not be contradicted by the EPS without defining what form the constraints take. In later sections, we will see that further attributes are necessary to define semantic composition with MRSs.

DEFINITION 3 (MRS Structure). An MRS structure is a tuple $\langle GT, R, C \rangle$ where GT is the top handle, R is a bag of EPS and C is a bag of handle constraints, such that:

Top: There is no handle h that outscopes GT .

For example, the MRS structure in (18) can be more precisely written as (19), where we have explicitly shown the top handle, $h0$, and the empty bag of handle constraints:

- (19) $\langle h0, \{h1: \text{every}(x, h2, h3), h2: \text{dog}(x), h4: \text{chase}(x, y),$
 $h5: \text{some}(y, h6, h7), h6: \text{white}(y), h6: \text{cat}(y)\}, \{\}\rangle$

We have also renumbered the handles and rearranged the EPS to be consistent with the convention we usually adopt of writing the EPS in the same order as the corresponding words appear in the sentence.

The definition of a scope-resolved MRS structure, which corresponds to a single expression in the object language, is now straightforward:

DEFINITION 4 (Scope-resolved MRS structure). A scope-resolved MRS structure is an MRS structure that satisfies the following conditions:

1. The MRS structure forms a tree of EP conjunctions, where dominance is determined by the outscopes ordering on EP conjunctions (i.e., a connected graph, with a single root that dominates every other node, and no nodes having more than one parent).
2. The top handle and all handle arguments (holes) are identified with an EP label.
3. All constraints are satisfied. With respect to outscopes constraints, for instance, if there is a constraint in C which specifies that E outscopes E' then the outscopes order between the EP conjunctions in R must be such that E outscopes E' .⁷

It follows from these conditions that in a scoped MRS structure all EP labels must be identified with a hole in some EP, except for the case of the top handle, which must be the label of the top node of the tree. The following example shows a scope-resolved MRS.

$$(20) \quad \langle h0, \{h0: \text{every}(x, h2, h3), h2: \text{dog}(x), h3: \text{sleep}(x)\}, \{\} \rangle$$

As we have seen, such an MRS can be thought of as a syntactic variant of an expression in the predicate calculus object language. For uniformity, we will actually assume it corresponds to a singleton set of object language expressions.

Notice that we have not said anything about the binding of variables, but we return to this at the end of this section. Also note that the definition of ‘immediately outscopes’ we have adopted is effectively syntactic, so the definition of a scope-resolved MRS is based on actual connections, not possible ones. Thus, the following MRS is not scope-resolved, even though there is trivially only one object-language expression to which it could correspond:

$$(21) \quad \langle h0, \{h0: \text{every}(x, h2, h3), h2: \text{dog}(x), h4: \text{sleep}(x)\}, \{\} \rangle$$

In order to formalize the idea that an underspecified MRS will correspond to a set of (potentially) more than one object-language expression, we will define the relationship of an arbitrary MRS structure to a set of scope-resolved MRSs. The intuition is that the pieces of the tree in an underspecified MRS structure may be linked up in a variety of ways to produce a set of maximally linked structures, where a maximally linked structure is a scope-resolved MRS. Linking simply consists of adding equalities between handles. Linking can be regarded as a form of specialization, and it is always monotonic.⁸ So if we have an MRS M that is equivalent to an MRS M' with the exception that M' contains zero or more additional equalities between handles, we can say that M *link-subsumes* M' . If $M \neq M'$ we can say that M strictly link-subsumes M' . For instance, the MRS shown in (19) (repeated as (22a) for convenience) strictly link-subsumes the structure shown in (22b) and is strictly link-subsumed by the structure in (22c):

$$(22) \quad \begin{array}{ll} \text{a. } \langle h0, \{h1: \text{every}(x, h2, h3), h2: \text{dog}(x), h4: \text{chase}(x, y), \\ h5: \text{some}(y, h6, h7), h6: \text{white}(y), h6: \text{cat}(y)\}, \{\} \rangle \\ \text{b. } \langle h0, \{h1: \text{every}(x, h2, h3), h2: \text{dog}(x), h3: \text{chase}(x, y), \\ h5: \text{some}(y, h6, h7), h6: \text{white}(y), h6: \text{cat}(y)\}, \{\} \rangle \\ \text{c. } \langle h0, \{h1: \text{every}(x, h2, h3), h2: \text{dog}(x), h4: \text{chase}(x, y), \\ h5: \text{some}(y, h6, h7), h8: \text{white}(y), h8: \text{cat}(y)\}, \{\} \rangle \end{array}$$

Interesting linkings involve one of three possibilities:

1. equating holes in EPs with labels of other EPs,
2. equating labels of EPs to form a larger EP conjunction, and
3. equating the top handle with the label of an EP conjunction.

It is also possible to equate two holes, but since such a structure is not a tree it cannot be a scope-resolved MRS, nor can it link-subsume a scope-resolved MRS. A scope-resolved structure is maximally linked in the sense that adding any further equalities between handles will give a non-tree structure.

Thus we have the following definition of a well-formed MRS (that is, one which will correspond to a non-empty set of object-language expressions):

DEFINITION 5 (Well-formed MRS structure). A well-formed MRS structure is an MRS structure that link-subsumes one or more scope-resolved MRSs.

We can say that if a well-formed MRS M link-subsumes a set of scope-resolved MRSs, $M_1, M_2, \dots, M_n$, such that M_1 corresponds to the (singleton) set of object-language expressions O_1 , and M_2 corresponds to O_2 and so on, then M corresponds to the set of object-language expressions which is the union of $O_1, O_2, \dots, O_n$. In general, the MRSs we are interested in when writing grammars are those that are well-formed by this definition.

Because we have not considered variable binding, this definition allows for partial MRSs, which will correspond to subsentential phrases, for instance. This is desirable, but we generally want to exclude specializations of complete MRSs (i.e., MRSs which correspond to sentences) which link the structure in such a way that variables are left unbound. In order to do this, we need to consider generalized quantifiers as being a special type of relation. An EP corresponding to a generalized quantifier has one variable argument, which is the bound variable of the quantifier, and two handle arguments, corresponding to the restriction and body arguments. If a variable does not occur as the bound variable of any quantifier, or occurs as the bound variable but also occurs outside the restriction or the body of that quantifier, then it is unbound. No two quantifiers may share bound variables. Thus a scope-resolved MRS without unbound variables will correspond to a closed sentence (a statement) in predicate calculus. The variable binding condition gives rise to outscopes constraints between quantifier EPs and EPs containing the bound variable in the obvious way.

This section has outlined how an MRS may be mapped into a set of object language expressions, but we also need to be able to go from an object language expression to a (scope-resolved) MRS expression. This is mostly straightforward, since it essentially involves the steps we discussed in the last section. However, one point we skipped over in the previous discussion is variable naming: although standard predicate calculus allows repeated variable names to refer to different variables when they are in the scope of different quantifiers, allowing duplicate variables in a single MRS gives rise to unnecessary complications which we wish to avoid, especially

when we come to consider feature structure representations. So, since there is no loss of generality, we assume distinct variables are always represented by distinct names in an MRS. When converting a predicate calculus formula to an MRS, variables must first be renamed as necessary to ensure the names are unique.

In order to go from a set of object language expressions to a single MRS structure, we have to consider the issue of representing explicit constraints on the valid linkings. The problem is not how to represent an arbitrary set of expressions, because we are only interested in sets of expressions that might correspond to the compositional semantics of individual natural language phrases. We discuss this in the next section.

4.2. CONSTRAINTS ON SCOPE RELATIONS

One major practical difference between underspecified representations is the form of constraints on scope relations they assume. We left this open in the preceding section, because MRS could potentially accommodate a range of possible types of constraints.⁹ Any form of scope constraints will describe restrictions on possible linkings of an MRS structure. As we have seen, there are implicit constraints which arise from the conditions on binding of variables by generalized quantifiers or from the tree conditions on the scope-resolved structures, and these are in fact enough to guarantee the example of scope resolution we saw in §3. But additional constraint specifications are required to adequately represent some natural language expressions.

For instance, although in example (19) we showed the restriction of the quantifiers being equated with a specific label, it is not possible to do this in general, because of examples such as (23a). If we assume *nephew* is a relational noun and has two arguments, it is generally accepted that (23a) has the two scopes shown in (23b,c):

- (23) a. Every nephew of some famous politician runs.
 b. $\text{every}(x, \text{some}(y, \text{famous}(y) \wedge \text{politician}(y), \text{nephew}(x, y)), \text{run}(x))$
 c. $\text{some}(y, \text{famous}(y) \wedge \text{politician}(y), \text{every}(x, \text{nephew}(x, y), \text{run}(x)))$
 d. $\langle h1, \{h2: \text{every}(x, h3, h4), h5: \text{nephew}(x, y), h6: \text{some}(y, h7, h8), h7: \text{politician}(y), h7: \text{famous}(y), h10: \text{run}(x)\}, \{\}$
 e. $\text{every}(x, \text{run}(x), \text{some}(y, \text{famous}(y) \wedge \text{politician}(y), \text{nephew}(x, y)))$
 f. $\text{some}(y, \text{famous}(y) \wedge \text{politician}(y), \text{every}(x, \text{run}(x), \text{nephew}(x, y)))$

This means the quantifier restriction must be underspecified in the MRS representation. However, if this is left unconstrained, as in, for instance, (23d), unwanted scopes can be obtained, specifically those in (23e,f), as well as the desirable scopes in (23b,c).¹⁰

There are a variety of constraints which we could use to deal with this, but in what follows, we will use one which was chosen because it makes

a straightforward form of semantic composition possible. We refer to this as a *qeq* constraint (or $=_q$) which stands for equality modulo quantifiers. A *qeq* constraint always relates a hole to a label. The intuition is that if a handle argument h is *qeq* to some label l , either that argument must be directly filled by l (i.e., $h=l$), or else one or more quantifiers ‘float in’ between h and l . In the latter case, the label of a quantifier fills the argument position and the body argument of that quantifier is filled either by l , or by the label of another quantifier, which in turn must have l directly or indirectly in its body.

An improved MRS representation of (23a) using *qeq* constraints is shown in (24):

- (24) $\langle h1, \{h2: \text{every}(x, h3, h4), h5: \text{nephew}(x, y), h6: \text{some}(y, h7, h8),$
 $h9: \text{politician}(y), h9: \text{famous}(y), h10: \text{run}(x)\},$
 $\{h1=_q h10, h7=_q h9, h3=_q h5\}\rangle$

Notice that there is a *qeq* relationship for the restriction of each quantifier and also one that holds between the top handle and the verb.

More formally:

DEFINITION 6 (qeq condition). An argument handle h is *qeq* to some label l just in case $h=l$ or there is some (non-repeating) chain of one or more quantifier EPS $E_1, E_2, \dots, E_n$ such that h is equal to the label of E_1 , l is equal to the body argument handle of E_n and for all pairs in the chain E_m, E_{m+1} the label of E_{m+1} is equal to the body argument handle of E_m .

The $=_q$ relation thus corresponds either to an equality relationship or a very specific form of outscopes relationship. Schematically, we can draw this as in Figure 1, where the dotted lines in the tree indicate a $=_q$ relation.

As a further example, consider the MRS shown in (25), where there is a scopal adverb, *probably*:

- (25) Every dog probably chases some white cat.
 $\langle h0, \{h1: \text{every}(x, h2, h3), h4: \text{dog}(x), h5: \text{probably}(h6), h7: \text{chase}(x, y),$
 $h8: \text{some}(y, h9, h10), h11: \text{white}(y), h11: \text{cat}(y)\},$
 $\{h0=_q h5, h2=_q h4, h6=_q h7, h9=_q h11\}\rangle$

This example is shown using the dotted line notation in Figure 2. This has the six scopes shown in (26).

- (26) a. $\text{probably}(\text{every}(x, \text{dog}(x), \text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{chase}(x, y))))$
 b. $\text{every}(x, \text{dog}(x), \text{probably}(\text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{chase}(x, y))))$
 c. $\text{every}(x, \text{dog}(x), \text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{probably}(\text{chase}(x, y))))$
 d. $\text{probably}(\text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{every}(x, \text{dog}(x), \text{chase}(x, y))))$
 e. $\text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{probably}(\text{every}(x, \text{dog}(x), \text{chase}(x, y))))$
 f. $\text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{every}(x, \text{dog}(x), \text{probably}(\text{chase}(x, y))))$

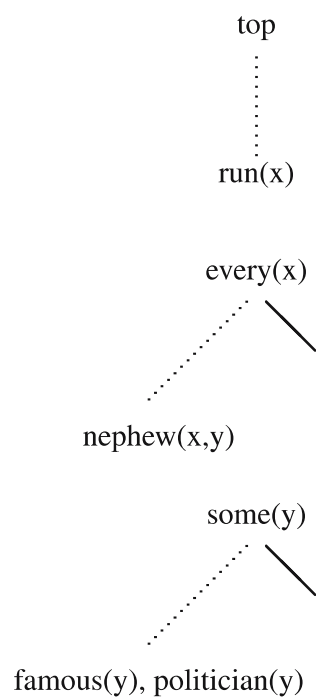


Figure 1. Schematic representation of the qeq relationships in *Every nephew of some famous politician runs.*

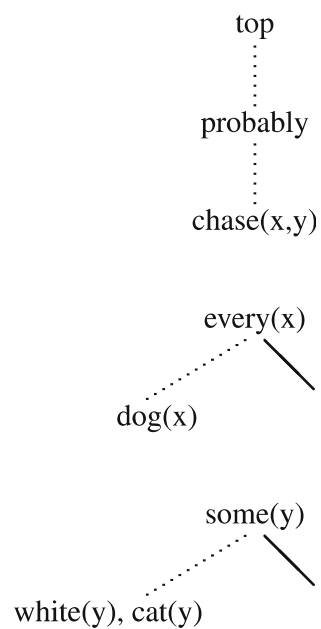


Figure 2. Schematic representation of the qeq relationships in *Every dog probably chases some white cat.*

Notice that, although *probably* is scopal, the scope cannot be resolved so that the EP for *probably* comes between two handles that are in a $=_q$ relationship. We say that the EP for *probably* is a fixed scopal EP, as opposed to *cat* (for instance), which has a non-scopal EP, and *every* and other quantifiers which have non-fixed (or floating) scopal EPs.

The point of using the qeq constraint is that it enables us to write simple rules of semantic composition (described in the next section), while directly and succinctly capturing the constraints on scope that otherwise could require mechanisms for quantifier storage. Using an outscopes constraint, rather than qeq, would potentially allow elementary predications to intervene other than quantifiers.¹¹

4.3. CONSTRAINTS ON HANDLES IN SEMANTIC COMPOSITION

In order to state the rules of combination involving handles, we will assume that MRSs have an extra attribute, the *local top*, or *ltop*, which will be the topmost label in an MRS which is not the label of a floating EP (i.e., not the label of a quantifier). For instance, in (25), the ltop is the label corresponding to the EP for *probably*, *h5*. In an MRS corresponding to a complete utterance, the top handle will be qeq to the local top. We will assume that when an MRS is composed, the top handle for each phrase in the composition is the same: that is, the top handle we introduced in §4.1 is a ‘global’ top handle *gtop*, in contrast to the local top for the phrase, which is the label of the topmost EP in that phrase which is not a floating EP. For phrases that only contain floating EPs, the ltop is not bound to any label. So the revised MRS structure is a tuple $\langle GT, LT, R, C \rangle$, where *LT* is the local top. As before, *GT* is the global top handle and *R* and *C* are the EP bag and handle constraints. For convenience, in the representations in this section, we will consistently use *h0* for the global top.

4.3.1. Lexical Items

We assume that each lexical item (other than those with empty EP bags) has a single distinguished main EP, which we will refer to as the *key* EP. All other EPs either share a label with the key EP or are equal to or qeq to some scopal argument of the key EP. With respect to key EPs there are three cases to consider:

1. The key is a non-scopal EP or a fixed scopal EP. In this case, the ltop handle of the MRS is equal to the label of the key EP. For example:¹²

$$\langle h0, h1, \{h1: \text{dog}(x)\}, \{\}, \langle h0, h2, \{h2: \text{probably}(h3)\}, \{\} \rangle$$

2. The key is a floating EP. In this case, the ltop handle of the MRS is not equated to any handle. For example:

$$\langle h0, h3, \{h4: \text{every}(y, h5, h6)\}, \{\}\rangle$$

3. If the lexical item has an empty EP bag there is no key EP and the lexical item's MRS structure has an unequated ltop handle. For example:

$$\langle h0, h1, \{\}, \{\}\rangle$$

4.3.2. Phrases

The bag of elementary predications associated with a phrase is constructed by appending the bags of EPS of all of the daughters. All handle constraints on daughters are preserved. The global top of the mother is always identified with the global top on each daughter. To determine the ltop of the phrase and any extra qeq constraints, we have to consider two classes of phrase formation, plus a root condition:

1. *Intersective combination*. The local tops of the daughters are equated with each other and with the local top of the phrase's MRS.

That is, if we assume the following:

mother MRS	$\langle GT_0, LT_0, R_0, C_0 \rangle$
daughter 1 MRS	$\langle GT_1, LT_1, R_1, C_1 \rangle$
...	
daughter n MRS	$\langle GT_n, LT_n, R_n, C_n \rangle$

then:

$$GT_0 = GT_1 = GT_2 \dots GT_n, LT_0 = LT_1 = LT_2 \dots LT_n,$$

$$R_0 = R_1 + R_2 \dots R_n, C_0 = C_1 + C_2 \dots C_n$$

(where + stands for append).

For example:

<i>white</i>	$\langle h0, h1, \{h1: \text{white}(x)\}, \{\}\rangle$
<i>cat</i>	$\langle h0, h2, \{h2: \text{cat}(y)\}, \{\}\rangle$
<i>whitecat</i>	$\langle h0, h1, \{h1: \text{cat}(x), h1: \text{white}(x)\}, \{\}\rangle$

Note that this class is not just intended for modification: it also covers the combination of a verb with an NP complement, for example.

2. *Scopal combination*. In the simplest case, this involves a binary rule, with one daughter containing a scopal EP which scopes over the other daughter. The daughter containing the scopal EP is the semantic head daughter. The handle-taking argument of the scopal EP is stated to be qeq to the ltop handle of the argument phrase. The ltop handle of the phrase is the ltop handle of the MRS which contains the scopal EP.

That is, if we assume the following:

mother MRS $\langle GT_0, LT_0, R_0, C_0 \rangle$
 scoping daughter MRS $\langle GT_s, LT_s, \{LT_s: EP(\dots, h) \dots\}, C_s \rangle$
 (where $EP(\dots, h)$ stands for an EP with a handle-taking argument)
 argument daughter MRS $\langle GT_{ns}, LT_{ns}, R_{ns}, C_{ns} \rangle$

then:

$$GT_0 = GT_s = GT_{ns}, LT_0 = LT_s, R_0 = R_s + R_{ns}, \\ C_0 = C_s + C_{ns} + \{h =_q LT_{ns}\}$$

For example:

sleeps $\langle h0, h5, \{h5: \text{sleep}(x)\}, \{\} \rangle$
probably $\langle h0, h2, \{h2: \text{probably}(h3)\}, \{\} \rangle$
probably sleeps $\langle h0, h2, \{h2: \text{probably}(h3), h5: \text{sleep}(x)\}, \{h3 =_q h5\} \rangle$

The extension for rules with multiple daughters where one scopal EP takes two or more arguments is straightforward: there will be a separate qeq constraint for each scopal argument.

For quantifiers, the scopal argument is always the restriction of the quantifier and the body of the quantifier is always left unconstrained.

For example:

every $\langle h0, h1, \{h2: \text{every}(x, h3, h4)\}, \{\} \rangle$
dog $\langle h0, h5, \{h5: \text{dog}(y)\}, \{\} \rangle$
every dog $\langle h0, h1, \{h2: \text{every}(x, h3, h4), h5: \text{dog}(x)\}, \{h3 =_q h5\} \rangle$

3. *Root*. The root condition stipulates that the global top is qeq to the local top of the root phrase. That is, if the MRS for the sentence is $\langle GT_0, LT_0, R_0, C_0 \rangle$ then C_0 must include $GT_0 =_q LT_0$.

Figure 3 shows in detail the composition of *every dog probably chases some white cat*. We have shown coindexation of the ordinary variables here as well as the handles, although we have not fully specified how this happens in the composition rules above since this requires discussion of the link to syntax. Here and below we will write ‘prbly’ rather than ‘probably’ as necessary to save space. As stated here, the constraints do not allow for any contribution of the construction itself to the semantics of the phrase. However, this is straightforwardly accommodated in MRS: the construction itself contributes semantics in exactly the same way as a daughter in a normal rule. Hence the construction may add one or more EPs that become part of the EP bag of the constructed phrase. Similarly a construction may add qeq constraints. Examples of this will be discussed in §6.6.

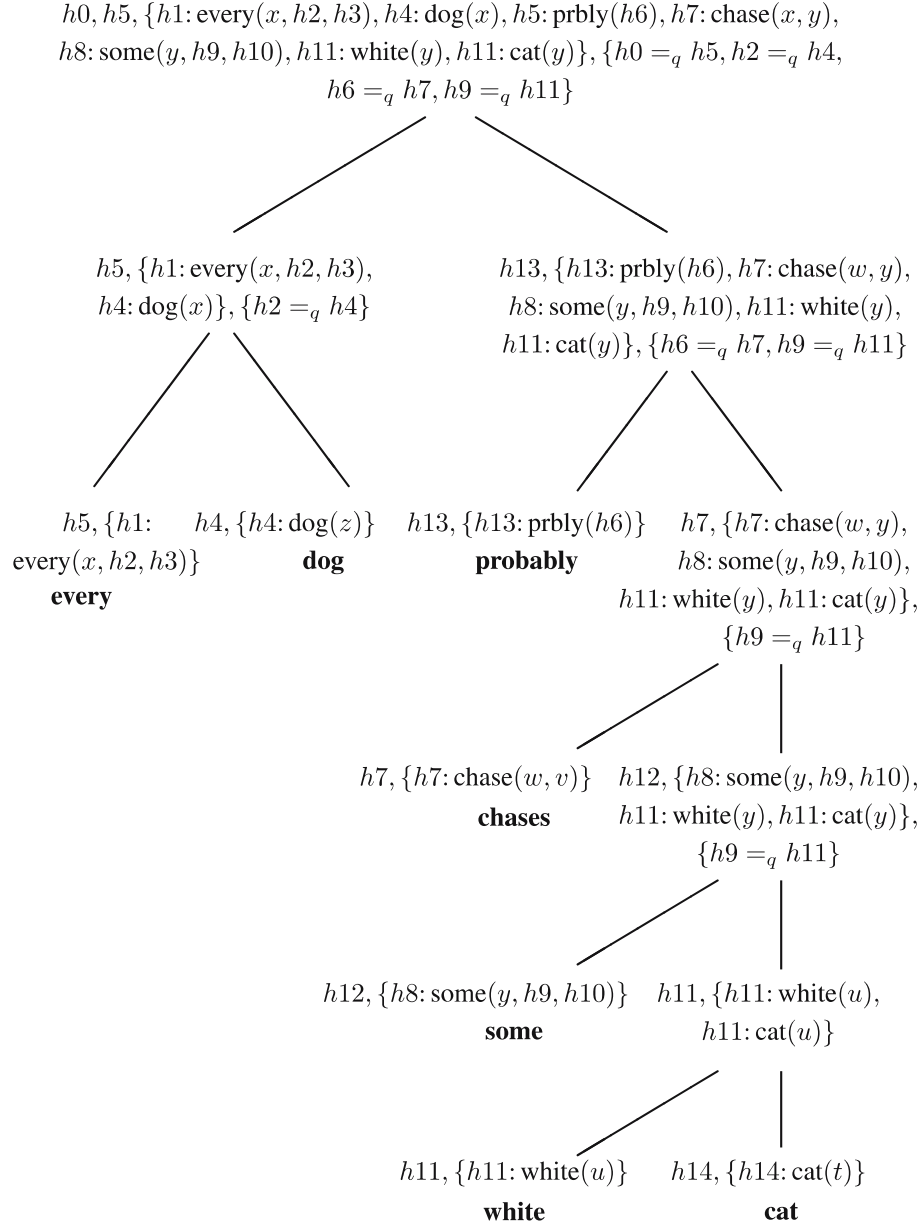


Figure 3. Example of composition. For reasons of space, empty sets of handle constraints are omitted and the global top handle, $h0$, is omitted except on the root.

The effect of the semantic composition rules stated above can be elucidated by the partial tree notation. Any MRS will have a backbone tree of non-floating scopal EPS, which are in a fixed relationship relative to each other, with the backbone terminated by non-scopal EPS. All quantifiers will

have a similar backbone associated with their restriction, but their body argument will be completely underspecified. All non-scopal EPs are in a *qeq* relationship, either with a quantifier's restriction or with an argument of a fixed-position scopal EP. Figures 1 and 2 from the previous section should help make this clear.

It should be clear that if a non-quantifier EP is not directly or indirectly constrained to be *qeq* to a scopal argument position or a quantifier's restriction, then the only way it can enter that part of the structure in a resolved MRS is if it is 'carried' there via the restriction of a quantifier. A quantifier in turn is restricted by the variable binding conditions and the requirement that it have a non-empty body. The combination of these conditions leads to various desirable properties, e.g., that nothing can get inside the restriction of a quantifier unless it corresponds to something that is within the corresponding noun phrase syntactically. For example, (27a) has the MRS shown in (27b), which has exactly the 5 scopes shown in (27c) to (27g):

- (27) a. Every nephew of some famous politician saw a pony.
 b. $\langle h0, h1, \{h2: \text{every}(x, h3, h4), h5: \text{nephew}(x, y), h6: \text{some}(y, h7, h8), h9: \text{politician}(y), h9: \text{famous}(y), h10: \text{see}(x, z), h11: a(z, h12, h13), h14: \text{pony}(z)\}, \{h0=_q h1, h1=_q h10, h7=_q h9, h3=_q h5, h12=_q h14\} \rangle$
 c. $\text{every}(x, \text{some}(y, \text{famous}(y) \wedge \text{politician}(y), \text{nephew}(x, y)), a(z, \text{pony}(z), \text{see}(x, z)))$
 d. $a(z, \text{pony}(z), \text{every}(x, \text{some}(y, \text{famous}(y) \wedge \text{politician}(y), \text{nephew}(x, y)), \text{see}(x, z)))$
 e. $\text{some}(y, \text{famous}(y) \wedge \text{politician}(y), \text{every}(x, \text{nephew}(x, y), a(z, \text{pony}(z), \text{see}(x, z))))$
 f. $\text{some}(y, \text{famous}(y) \wedge \text{politician}(y), a(z, \text{pony}(z), \text{every}(x, \text{nephew}(x, y), \text{see}(x, z))))$
 g. $a(z, \text{pony}(z), \text{some}(y, \text{famous}(y) \wedge \text{politician}(y), \text{every}(x, \text{nephew}(x, y), \text{see}(x, z))))$

We leave it as an exercise for the reader to work through the representations to see why only these five scopes are valid. This essentially corresponds to the effects of quantifier storage when used with generalized quantifiers.¹³

Although it is often assumed that constraints on quantifier scoping are parallel to constraints on filler-gap dependencies, there is considerable evidence that this is not the case. For example:

- (28) a. We took pictures of Andy and every one of his dance partners.
 (violates Coordinate Structure Constraint)
 b. Someone took a picture of each student's bicycle. (violates Left Branch Condition, however analyzed)

- c. We were able to find someone who was an expert on each of the castles we planned to visit. (crossed scope reading violates Complex NP Constraint, however analyzed)

Constraints on quantifier scoping, which involve lexical grading (e.g., *each* scopes more freely than *all*, which in turn scopes more freely than *no*), are, in our view, largely reducible to general considerations of processing complexity.

The approach described here means that there can be no ambiguity involving the relative scope of two non-quantifiers unless there is lexical or syntactic ambiguity. For instance, the two readings of (29a), sketched in (29b), have to arise from a syntactic ambiguity.

- (29) a. Kim could not sleep.
- b. could(not(sleep(kim)))
- not(could(sleep(kim)))

This seems to be justified in this particular case (see e.g., Warner 2000), but there are other phenomena that may require these conditions to be relaxed. However, we will not discuss this further in this paper.

5. MRS in Typed Feature Structures

As stated in the introduction, we are using MRS for grammars developed within a typed feature structure formalism. In this section we will briefly reformulate what has already been discussed in terms of typed feature structures, describing some implementation choices along the way. This encoding is formalism-dependent but framework-independent: it is equally applicable to categorial grammars encoded in typed feature structures, for instance, although some of our encoding choices are motivated by following the existing HPSG tradition. In the next section we will point out various descriptive advantages of MRS as we move to a discussion of its use in a particular grammar, namely the English Resource Grammar (ERG) developed by the LinGO project at CSLI (Flickinger et al. 2000; see also <http://www.delph-in.net/>, from which the grammar itself can be downloaded, along with various software).

5.1. BASIC STRUCTURES

In feature structure frameworks, the semantic representation is one piece of a larger feature structure representing the word or phrase as a whole. Here we will concentrate on the semantic representations, ignoring links with the syntax.

First we consider the encoding of EPS. We will assume that the EP's relation is encoded using the feature structure's type, and that there is one feature corresponding to the handle that labels the EP and one feature for each of the EP's argument positions. For example, (30) illustrates the feature structure encoding of the EPS introduced by the lexical entries for *dog* and *every*:

- (30) a.
$$\begin{bmatrix} \text{dog_rel} \\ \text{LBL handle} \\ \text{ARG0 ref-ind} \end{bmatrix}$$
- b.
$$\begin{bmatrix} \text{every_rel} \\ \text{LBL handle} \\ \text{ARG0 ref-ind} \\ \text{RSTR handle} \\ \text{BODY handle} \end{bmatrix}$$

By convention, the types corresponding to relations which are introduced by individual lexemes have a leading underscore followed by the orthography of the lexeme. Different senses may be distinguished, but we omit these distinctions here for purposes of readability. All types denoting relations end in ‘_rel’. As usual in typed feature structure formalism encodings of semantics, coindexation is used to represent variable identity; we take a similar approach with handles. Notice that there is no type distinction between handles in label position and argument handles. This allows the linking of an MRS to be defined in terms of the feature structure logic, since adding links corresponds to adding reentrancies/coindexations.

We should mention that we avoid the use of very specific features for relations, such as *CORPSE* for the subject of *die*. This use of such features is traditional in the HPSG literature, but it is incompatible with stating generalizations involving linking between syntax and semantics in lexical entries. For instance, separate linking rules would have to be stated for *die* and *snore* declaring the reentrancy of the subject's index with the relevant attribute in the verb's relation. We therefore assume there is a relatively small set of features that are appropriate for the different relations. The general MRS approach is neutral about what this inventory of relation features consists of, being equally compatible with the use of generalized semantic (thematic) roles such as *ACTOR* and *UNDERGOER* (e.g., following Davis, 2001) or a semantically-bleached nomenclature, such as *ARG1*, *ARG2*. We will adopt the latter style here, to maintain consistency with usage in the ERG.

An MRS is defined as corresponding to a feature structure of type **mrs** with features *HOOK*, *RELS* and *HCONS*, where the value of *HOOK* is of type **hook**, which itself defines (at least) the features *GTOP* and *LTOP*. *GTOP* introduces the global top handle, *LTOP* the local top. *HOOK* is used to group together the features that specify the parts of an MRS which are visible to semantic functors, following the approach to semantic composition in Copestake et al., (2001). Note that *HOOK* also includes the attribute *INDEX*, which corresponds to a distinguished normal (non-handle) variable, as illustrated below. *RELS* is the

feature that introduces the bag of EPS, which is implemented as a list in the feature structure representation.¹⁴ HCONS introduces the handle constraints, which are also implemented as a list. The individual qeq constraints are represented by a type **qeq** with appropriate features HARG for the hole and LARG for the label.

The MRS feature structure for the sentence *every dog probably sleeps* is shown along with its non-feature-structure equivalent in (31):

$$(31) \quad \left[\begin{array}{l} \text{mrs} \\ \text{HOOK} \left[\begin{array}{l} \text{hook} \\ \text{GTOP} \boxed{1} \text{ handle} \\ \text{LTOP} \boxed{7} \text{ handle} \end{array} \right] \\ \text{RELS} < \left[\begin{array}{l} \text{.every_rel} \\ \text{LBL} \boxed{2} \text{ handle} \\ \text{ARG0} \boxed{3} \text{ ref-ind} \\ \text{RSTR} \boxed{4} \text{ handle} \\ \text{BODY} \text{ handle} \end{array} \right], \left[\begin{array}{l} \text{.dog_rel} \\ \text{LBL} \boxed{6} \text{ handle} \\ \text{ARG0} \boxed{3} \end{array} \right], \left[\begin{array}{l} \text{.prbly_rel} \\ \text{LBL} \boxed{7} \text{ handle} \\ \text{ARG1} \boxed{8} \end{array} \right], \left[\begin{array}{l} \text{.sleep_rel} \\ \text{LBL} \boxed{9} \\ \text{ARG1} \boxed{3} \end{array} \right] > \\ \text{HCONS} < \left[\begin{array}{l} \text{qeq} \\ \text{HARG} \boxed{1} \\ \text{LARG} \boxed{7} \end{array} \right], \left[\begin{array}{l} \text{qeq} \\ \text{HARG} \boxed{4} \\ \text{LARG} \boxed{6} \end{array} \right], \left[\begin{array}{l} \text{qeq} \\ \text{HARG} \boxed{8} \\ \text{LARG} \boxed{9} \end{array} \right] > \end{array} \right]$$

$$\{h1, h7, \{h2: \text{every}(x, h4, h5), h6: \text{dog}(x), h7: \text{prbly}(h8), h9: \text{sleep}(x)\},$$

$$\{h1 =_q h7, h4 =_q h6, h8 =_q h9\}\}$$

For ease of comparison, the handle numbering has been made consistent between the two representations, though there is no number for the BODY of the relation for *every* in the feature structure representation, because there is no coindexation. Conversely, the variable which is represented as x in the non-feature-structure form corresponds to the coindexation labeled with $\boxed{3}$ in the feature structure form.

There are two alternative representation choices which we will mention here, but not discuss in detail. The first is that we could have chosen to make the relation be the value of a feature (such as RELN) in the EP, rather than the type of the EP as a whole, as in Pollard and Sag (1994). This is shown in (32):

$$(32) \quad \left[\begin{array}{l} \text{RELN} \text{ .dog_rel} \\ \text{LBL} \text{ handle} \\ \text{ARG0} \text{ ref-ind} \end{array} \right]$$

This allows the possibility of complex relations such as the equivalent of *very(big)(x)*, and reduces the number of distinct types required in a grammar with a large lexicon, but would slightly complicate the presentation here. Second, instead of treating scopal arguments via handle identity, as we have been assuming, it would be possible to get a similar effect without the LBL (label) feature, by making use of coindexation with the whole EP. This is only possible, however, if we use an explicit relation for conjunction, instead of relying on handle identity.

5.2. COMPOSITION

We can straightforwardly reformulate the principles of semantic composition given in §4.3 in terms of feature structures. The classes of EP (i.e., fixed scopal, non-scopal, etc.) can be distinguished in terms of their position in the type hierarchy. The semantics for the lexical entry for *dog*, *probably* and *every* with the hook features appropriately coindexed with the elementary predications are shown in (33):

$$(33) \quad \begin{array}{l} \text{dog} \\ \text{probably} \\ \text{every} \end{array} \left[\begin{array}{l} \text{mrs} \\ \text{HOOK} \left[\begin{array}{l} \text{GTOP handle} \\ \text{LTOP } \boxed{2} \\ \text{INDEX } \boxed{3} \end{array} \right] \\ \text{RELS} < \left[\begin{array}{l} \text{.dog_rel} \\ \text{LBL } \boxed{2} \\ \text{ARG0 } \boxed{3} \end{array} \right] > \\ \text{HCONS} < > \end{array} \right]$$

$$\left[\begin{array}{l} \text{mrs} \\ \text{HOOK} \left[\begin{array}{l} \text{GTOP handle} \\ \text{LTOP } \boxed{2} \end{array} \right] \\ \text{RELS} < \left[\begin{array}{l} \text{.probably_rel} \\ \text{LBL } \boxed{2} \\ \text{ARG1 } \boxed{3} \end{array} \right] > \\ \text{HCONS} < \left[\begin{array}{l} \text{qeq} \\ \text{HARG } \boxed{3} \\ \text{LARG handle} \end{array} \right] > \end{array} \right]$$

$$\left[\begin{array}{l} \text{mrs} \\ \text{HOOK} \left[\begin{array}{l} \text{GTOP handle} \\ \text{LTOP handle} \\ \text{INDEX } \boxed{3} \end{array} \right] \\ \text{RELS} < \left[\begin{array}{l} \text{.every_rel} \\ \text{LBL handle} \\ \text{ARG0 } \boxed{3} \\ \text{RSTR } \boxed{4} \\ \text{BODY handle} \end{array} \right] > \\ \text{HCONS} < \left[\begin{array}{l} \text{qeq} \\ \text{HARG } \boxed{4} \\ \text{LARG handle} \end{array} \right] > \end{array} \right]$$

Notice that the INDEX is equated with the ARG0 in both *every* and *dog*. The LBL of *.every_rel* is not coindexed with the LTOP, in accordance with the rules in §4.3. We have not shown the link to the syntactic part of the sign, but this would cause the LARG of the qeq in *probably* to be coindexed with the ltop of the structure it modifies. Similarly, the LARG of the qeq in *every* would be coindexed with the ltop of the Nbar, while the ARG0 would be coindexed with the Nbar's index. As discussed in detail in Copestake et al. (2001), MRS composition can be formalised in terms of a semantic head daughter having a particular syntactic *slot*¹⁵ which is filled by the hook of the non-semantic-head daughter: we will return to this in §6.2.

The simple type hierarchy shown in Figure 4 sketches one way of encoding the constraints on scopal and intersective phrases. In these types, the semantic head daughter is DTR1 and the other daughter is DTR2. The idea is that the hook of the mother is always the hook of the semantic head daughter, that GTOPs are always equated, and RELS and HCONS attributes are always accumulated. This figure incorporates several simplifying assumptions (for expository convenience):

1. It assumes all phrases are binary (though the extension to unary and ternary rules is straightforward).
2. The constraint for scopal phrases is very schematic, because it actually has to generalise over a range of syntactic constructions.
3. The appending of the accumulator attributes RELS and HCONS can be implemented by difference lists in those variants of feature structure formalisms which do not support an append operator directly.
4. Not all constructions of a grammar need to be either intersective or scopal inherently, since in a highly lexicalist framework some constructions, such as the rule(s) for combining a lexical head with its complements, might well inherit from the more general (underspecified) type, leaving the lexical head to determine the form of composition. However, each instantiation of a rule in a phrase will be intersective or scopal.

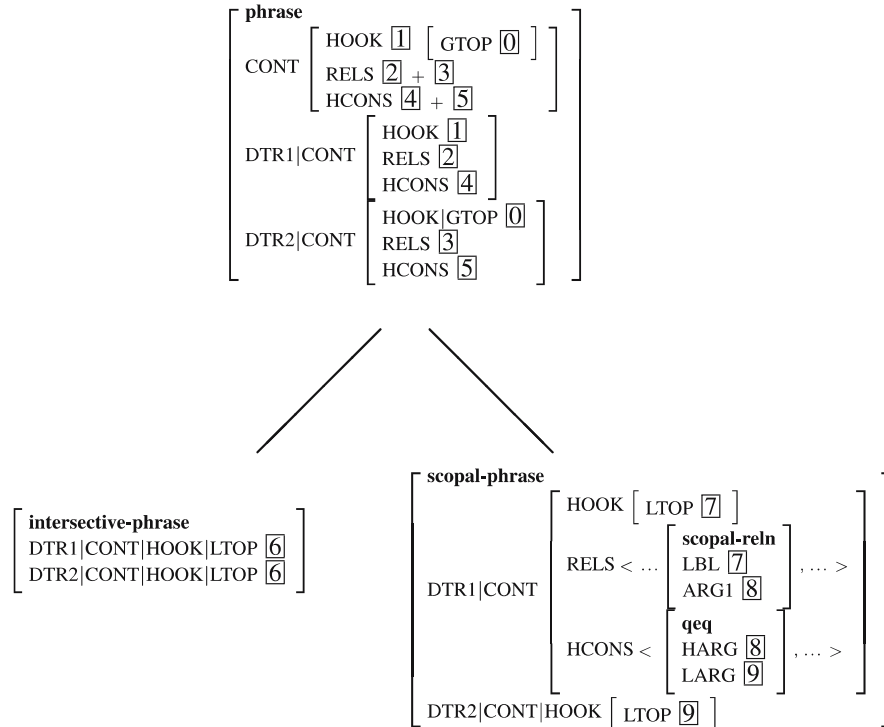


Figure 4. Illustrative constraints on composition.

Using these definitions of semantic composition in feature structures, the composition of the sentence *Every dog probably sleeps* is shown in Figure 5. Notice that since we are now talking about feature structures and unification, we have illustrated how the coindexation propagates through the structure in this figure. In addition, we represent the root condition as an extra level in the tree; this is not essential, but provides a locus for the constraint identifying the highest *LTOP* of the sentence with the value of *GTOP*.

In this example, notice that the lexical entries for the quantifier *every* and the scopal adverb *probably* each introduce a **qeq** constraint which is then passed up through the tree by virtue of the simple append of *HCONS* values included in the principles of semantic composition. Further, we only illustrate here lexical entries which each introduce a single *EP* but a grammar may well include semantically complex lexical entries which introduce two or more relations. In such cases, we require that the semantics of these semantically complex lexical entries conform to the well-formedness principles already defined for phrases, though we may also need an additional type of handle constraint for simple equality. To see this, consider the example *non-recursive*: if prefixation by *non-* is treated as productive, the semantics for this can be treated as consisting of two relations, where the **_non_rel** immediately outscopes the **_recursive_rel**.

5.3. RESOLUTION

MRS was designed so that the process of scope resolution could be formulated as a monotonic specialization of a feature structure (see Nerbonne, 1993). It is thus consistent with the general constraint-based approach assumed in HPSG. The discussion of linking of tree fragments in §4 holds in much the same way for feature structures: linking consists of adding a coindexation between two handles in the feature structure, and the link-subsumption relationship described between MRSs is thus consistent with the partial order on feature structures, though the feature structure partial order is more fine-grained.

In principle, it would be possible to implement the instantiation of an underspecified MRS to produce scoped forms directly in a typed feature structure formalism that allowed recursive constraints. However, to do this would be complicated and would raise very serious efficiency issues. The number of scopes of a sentence with *n* quantifiers and no other scopal elements is *n!* (ignoring complications, e.g., one quantifier's scope being included in the restriction of another) and a naive algorithm is therefore problematic. Because of this, we have developed external modules for extra-grammatical processing such as scope determination. The situation is analogous to parsing and generation using typed feature structures. It is possible to describe a grammar in a typed feature structure

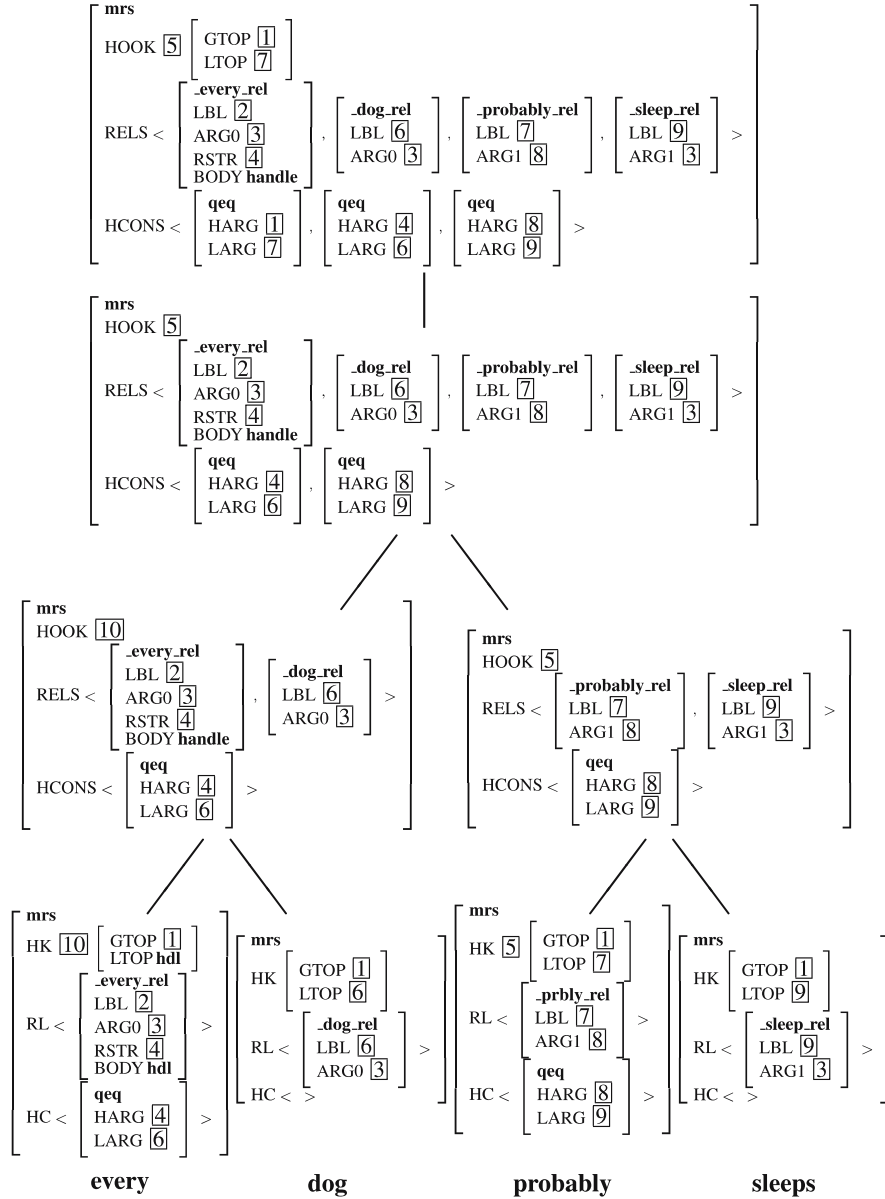


Figure 5. Example of composition using feature structures.

formalism in such a way that parsing and generation can be regarded as a form of resolution of recursive constraints. However, for the sake of efficiency, most systems make use of special-purpose processing algorithms, such as variants of chart-parsing, rather than relying on general constraint resolution. And just as the use of a chart can avoid exponentiality in

parsing, non-factorial algorithms can be designed for determining whether an MRS expressed using dominance constraints has at least one scope (see Niehren and Thater, 2003 and other work on the CHORUS project). However, we will not discuss this further in this paper.

6. Using MRS in an HPSG

In this section we describe the use of MRS in a large HPSG, namely the LinGO project's English Resource Grammar (Flickinger et al., 2000¹⁶). We will do this through a series of examples, beginning with an amplification of one we have discussed already, and then going through a series of somewhat more complex cases.

6.1. A BASIC EXAMPLE

First, consider (34), which is the representation of the sentence *Every dog probably slept* produced by the LinGO ERG:¹⁷ (This structure should be compared with (31).)

$$(34) \left[\begin{array}{l} \text{mrs} \\ \text{HOOK} \left[\begin{array}{l} \text{hook} \\ \text{INDEX } [0] \left[\begin{array}{l} \text{event} \\ \text{TENSE } \text{past} \\ \text{MOOD } \text{indicative} \end{array} \right] \\ \text{LTOP } [1] \text{ handle} \end{array} \right] \\ \text{RELS} < \left[\begin{array}{l} \text{every_rel} \\ \text{LBL } [2] \\ \text{ARG0 } [3] \left[\begin{array}{l} \text{ref-ind} \\ \text{PN } \text{3sg} \\ \text{GEN } \text{nt} \end{array} \right] \\ \text{RSTR } [4] \text{ handle} \\ \text{BODY } \text{handle} \end{array} \right], \left[\begin{array}{l} \text{dog_rel} \\ \text{LBL } [6] \\ \text{ARG0 } [3] \end{array} \right], \left[\begin{array}{l} \text{prbly_rel} \\ \text{LBL } [7] \\ \text{ARG1 } [8] \end{array} \right], \left[\begin{array}{l} \text{sleep_rel} \\ \text{LBL } [9] \\ \text{ARG0 } [0] \\ \text{ARG1 } [3] \end{array} \right], \left[\begin{array}{l} \text{prpstn_m_rel} \\ \text{LBL } [1] \\ \text{ARG0 } [0] \\ \text{MARG } [10] \end{array} \right] > \\ \text{HCONS} < \left[\begin{array}{l} \text{qeq} \\ \text{HARG } [10] \\ \text{LARG } [7] \end{array} \right], \left[\begin{array}{l} \text{qeq} \\ \text{HARG } [4] \\ \text{LARG } [6] \end{array} \right], \left[\begin{array}{l} \text{qeq} \\ \text{HARG } [8] \\ \text{LARG } [9] \end{array} \right] > \end{array} \right]$$

The basic structures are as described in the previous section: the overall structure is of type **mrs** and it has features **HOOK**, **RELS** and **HCONS** as before. However, the global **GTOP** is omitted, as it is redundant for this grammar (as we will explain shortly).

6.1.1. Complex Indices

In the previous section, we assumed that the variables were represented by coindexed atomic values. In the LinGO ERG, as is usual in HPSG, these variables have their own substructure, which is used to encode attributes for agreement (Pollard and Sag, 1994; chapter 2) or tense and aspect. The details of the encoding are not relevant to the MRS representation, however.

6.1.2. *Event Variables*

In this example, the value of the `INDEX` attribute is an event variable, which is coindexed with the `ARG0` attribute of the `_sleep_rel` relation (and also with the `prpstn_m_rel`). The ERG, in common with much other work on computational semantics, uses a form of Davidsonian representation which requires that all verbs introduce events. One effect of this is that it leads to a situation where there are two different classes of adverbs: scopal adverbs such as *probably*, which are treated as taking arguments of type **handle** in MRS, and non-scopal adverbs, such as *quickly*, which are treated as taking events as arguments. This is analogous to the distinction between non-intersective and intersective adjectives.

Notice that the event variables are not explicitly bound. We assume that there is an implicit wide-scope quantifier for each event variable. This is not entirely adequate, because there are some examples in which the scope of events could plausibly be claimed to interact with the explicit quantifiers, but we will not pursue that issue further here.

6.1.3. *Tense*

As can be seen in this example, the grammar encodes tense and mood by instantiating values on the event variable of the verb. Effectively these values simply record the information derived from the morphology of a verb (or an auxiliary). However, enough information is present to allow conversion to a more complex form if necessary. For instance, assume that the effect of the `TENSE` feature having the value **past** is cached out in a conventional logical representation as $\text{past}(e)$. A more semantically interesting representation might be $\text{hold}(e, t)$, $\text{precedes}(t, \text{now})$, where *hold* and *precedes* are primitives in the logic of events and times, and *now* is a variable representing the sentence time. But this representation can be derived from the simpler one given, since the *now* of the sentence, although semantically a variable, is unaffected by the sentence content. Similarly, the only point of explicitly using the additional temporal variable *t* in the grammar would be if it turns out to be necessary to refer to it directly in some other part of the semantics. It might seem that a temporal adverbial, such as *on Sunday* should refer to *t*, but even if this is correct, it does not follow that it is necessary to make the `_on_rel` take *t* as an argument rather than the event *e*, since there is a function from *e* to *t*. Furthermore, the grammar is significantly simpler if `_on_rel` takes an event argument like other adverbial prepositions. We have belabored this point somewhat, because it is representative of a more general issue concerning semantics in a computational grammar: simple, generic-looking structures are preferable for processing, maintainability, and comprehensibility, provided that enough information is included for a more explicit representation to be derived straightforwardly.

6.1.4. *Message Relations*

The example shown in (34) contains an ‘extra’ EP (compared to (33)) which has a relation **prpstn.m.rel**. This conveys the information that the sentence is a proposition, rather than a question, for instance (questions are indicated by **int.m.rel**). Such relations, generically referred to as *message* relations, are an adaptation of the Vendlerian theory of messages developed by Ginzburg and Sag (2000).

In general, message relations behave like normal scopal relations with respect to their arguments (introduced with the attribute **MARG** for message argument), but they are a special case when they are themselves arguments of some sort of scopal EP, since they are always equal to the scopal EP’s argument position rather than being **qeq** to it. This avoids our use of message relations introducing additional scope possibilities. The root condition is also modified in a similar way, so that the global top of the MRS is equal to rather than **qeq** to the label of the message relation. Since all MRSs corresponding to root structures have an outermost scoping message, this has the effect that all the quantifiers in the sentence have to scope under the outermost message, here **prpstn.m.rel**. In fact this means that the feature **GTOP** is redundant in this grammar, since the **LTOP** of a root structure is also the global top.¹⁸

6.2. COMPOSITION IN THE ERG

The principles of composition in the ERG are consistent with the approach described by Copestake et al. (2001). To describe this informally, we will need the concept of a slot (misleadingly called ‘hole’ by Copestake et al.): a semantic argument position in a word or phrase A that is associated with syntactic constraints on the word or phrase B whose semantics will supply that argument when the relevant grammar rule combines A and B. Slots represent the syntax-semantics link: they control how variables in the EPs in the semantic head daughter (with which the slot is associated) get identified with one or more attributes of the hook of the non-head-daughter. For instance, in a head-modifier rule, the semantic head daughter is the modifier, and the slot corresponds to the **MOD** feature. The mod slot of the modifier is identified with the hook of the modified element, constraining its **LTOP** attribute and (for intersective modifiers) its **INDEX**. In the ERG, the slot mechanism is implemented by typed feature structures, as is usual in HPSG. However, slots are a convenient generalization that enable us to describe semantic composition without having to discuss the feature structure architecture and which can be implemented in non-feature structure formalisms.

The general principles of semantic composition can be stated as follows:

1. The **RELS** value of a phrase is the concatenation (append) of the **RELS** values of the daughters.

2. The `HCONS` of a phrase is the concatenation (append) of the `HCONS` values of the daughters.
3. The `HOOK` of a phrase is the `HOOK` of the semantic head daughter, which is determined uniquely for each construction type.
4. One slot of the semantic head daughter of a phrase is identified with the hook in the other daughter. For scopal combination, this will lead to a `qeq` relation between the scopal argument in an `EP` and the `LTOP` of that other daughter. For intersective combination, an argument in an `EP` will be identified with the `INDEX` of that other daughter, and the `LTOP` of that daughter will be identified with the `LTOP` of the semantic head.

These rules need some elaboration for the `ERG`. One complication is that it is possible for phrases to themselves introduce semantic content directly. This is done via the `C-CONT` (`CONSTRUCTIONAL-CONTENT`) attribute of the phrase (an example is discussed below in §6.6). However, the `C-CONT` attribute's value meets the constraints on lexical structures that have previously been described and `C-CONT` behaves for purposes of composition just like the semantics of an additional daughter. It is possible for the `C-CONT` to be the semantic head of a phrase.

6.3. INTERSECTIVE AND SCOPAL MODIFIERS

Bob Kasper, in an unpublished paper from 1996, investigates the internal semantic structure of modifiers, observing that previous analyses of modification in `HPSG` failed to adequately account for modifier phrases that contain modifiers of their own, as in his example (35):

(35) Congress approved the potentially controversial plan.

He notes that in the analysis of modifier semantics given in Pollard and Sag (1994), the argument of *potentially* ends up including not only the `psoa` for the adjective *controversial* but also the `psoa` for *plan*. This analysis predicts incorrectly that in the above example the plan itself is only a potential one, which means that it fails to predict, for example, that a sentence like (35) entails that Congress approved a plan.

We can also describe the problem in terms of `MRS` semantics. In `HPSG`, the syntactic feature `MOD` is used to allow a modifier to specify the kind of sign that it can modify. Thus the `MOD` value of an adjective phrase is instantiated by the `synsem` of the noun it modifies. In recursive modification, the `MOD` feature of the phrase is identified with that of the modified structure: for instance, the `MOD` feature for *potentially controversial* is identified with that of *controversial*. This leads to the semantic problem: if an adjective is intersective, like *controversial*, the `LTOP` of the adjective and the noun should be equated in a phrase like *the controversial plan*. But if this is achieved via a `MOD` specification in the lexical entry of the adjective, then

the result is wrong for *the potentially controversial plan*, since this would mean that *plan* and *controversial* share a label and thus when *potentially* is given scope over the EP for *controversial* it would also end up with scope over the EP for *plan*.

Kasper proposes that this and related difficulties with the original analysis of modifier semantics can be rectified by distinguishing the inherent semantic content of a modifier from the combinatorial semantic properties of the modifier. Enriching the MOD feature to include attributes ICONT (internal content) and ECONT (external content), he shows how this distinction can be formalized to produce a more adequate account of recursive modifier constructions while maintaining the semantic framework of Pollard and Sag.

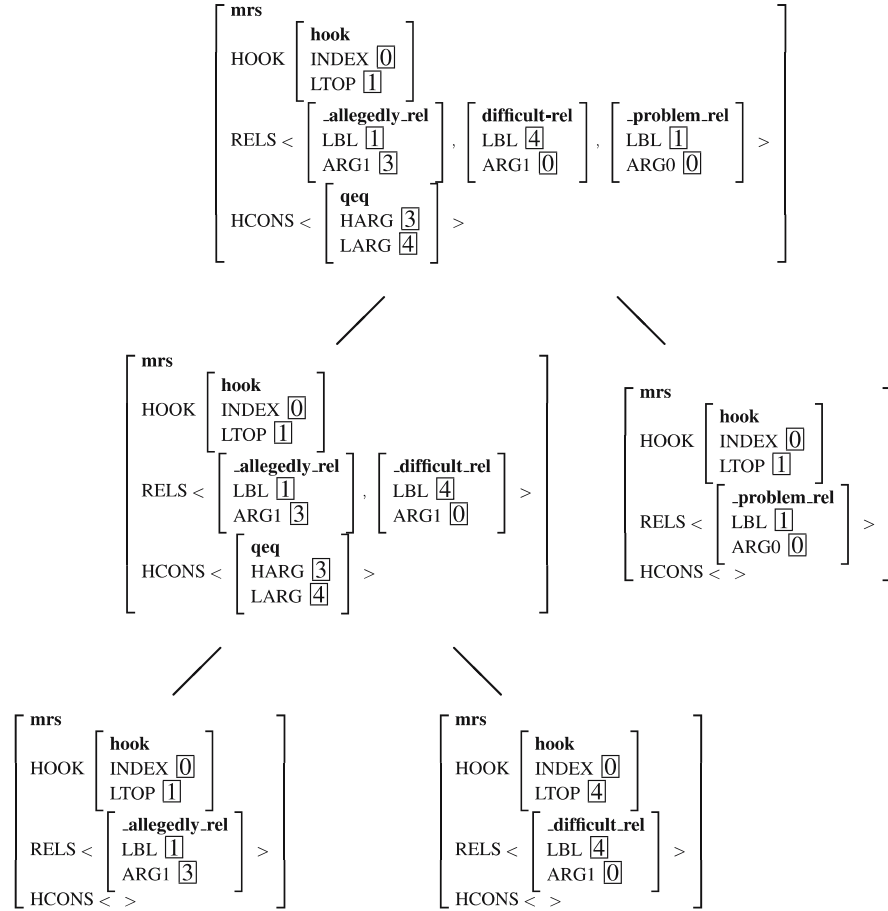
Kasper's account could be reformulated in MRS, but we provide an alternative which captures the correct semantics without having to draw the internal/external content distinction. Unlike Kasper, we do have to assume that there are two rules for modification: one for scopal and one for non-scopal modifiers. But this distinction has fewer ramifications than Kasper's. The crucial difference between the rules is that the intersective modifier rule equates the LTOP values of the two daughters, while the scopal version does not. This removes the need for coindexation of ltops in the the lexical entries of intersective adjectives and thus avoids Kasper's problem. In Figure 6 we show the composition of the MRS for *allegedly difficult problem*.

6.4. COMPLEMENTS AND SCOPE

Verbs which take sentential complements are analyzed as handle-taking in MRS in much the same way as scopal adverbs. This is illustrated in (36) below: for simplicity, we ignore the message relation in this section.

$$(36) \left[\begin{array}{l} \text{PHON } \text{thinks} \\ \text{HEAD } \text{verb} \\ \text{ARG-ST } < \left[\begin{array}{l} \text{CAT } NP \\ \text{HOOK|INDEX } \boxed{1} \end{array} \right], \left[\begin{array}{l} \text{CAT } S \\ \text{HOOK|LTOP } \boxed{4} \end{array} \right] > \\ \text{CONT } \left[\begin{array}{l} \text{mrs} \\ \text{HOOK|LTOP } \boxed{0} \\ \text{RELS } < \left[\begin{array}{l} \text{think_rel} \\ \text{LBL } \boxed{0} \\ \text{ARG1 } \boxed{1} \\ \text{ARG2 } \boxed{2} \end{array} \right] > \\ \text{HCONS } < \left[\begin{array}{l} \text{qeq} \\ \text{HARG } \boxed{2} \\ \text{LARG } \boxed{4} \end{array} \right] > \end{array} \right] \end{array} \right]$$

Note that the scope of quantifiers within complement clauses is unconstrained, as in the three-way scope ambiguity (at least) of examples like (37):

Figure 6. MRS for *allegedly difficult problem*.

(37) Everyone thought Dana liked a friend of mine from East Texas.

However, the scope of an adverb within a complement clause is bounded by that clause's ltop, as discussed earlier. Hence we correctly predict that adverbs like *probably* in examples like (38) are unambiguously within the scope of *_think_rel*:

(38) Everyone thought Dana probably left.

Note further that the famed scope ambiguity of examples like (39) is properly handled within MRS:

(39) A unicorn appears to be approaching.

The lexical analysis of raising commonly assumed within HPSG identifies the index bound by the existential quantifier with that of the unexpressed

subject of the VPs *appears to be approaching*, *to be approaching*, *be approaching* and *approaching*. Given this, it follows that this index is also the argument of **_approach_rel**. Thus, the MRS for a sentence like (39) is as described in (40):

$$(40) \left[\begin{array}{l} \text{mrs} \\ \text{HOOK|LTOP } \boxed{1} \text{ hndl} \\ \text{RELS} < \left[\begin{array}{l} \text{a_q_rel} \\ \text{LBL hndl} \\ \text{ARG0 } \boxed{3} \\ \text{RSTR } \boxed{4} \text{ hndl} \\ \text{BODY hndl} \end{array} \right], \left[\begin{array}{l} \text{unicorn_rel} \\ \text{LBL } \boxed{6} \\ \text{ARG0 } \boxed{3} \end{array} \right], \left[\begin{array}{l} \text{appear_rel} \\ \text{LBL } \boxed{7} \\ \text{ARG1 } \boxed{8} \text{ hndl} \end{array} \right], \left[\begin{array}{l} \text{approach_rel} \\ \text{LBL } \boxed{9} \\ \text{ARG1 } \boxed{3} \end{array} \right] > \\ \text{HCONS} < \left[\begin{array}{l} \text{qeq} \\ \text{HARG } \boxed{1} \\ \text{LARG } \boxed{7} \end{array} \right], \left[\begin{array}{l} \text{qeq} \\ \text{HARG } \boxed{4} \\ \text{LARG } \boxed{6} \end{array} \right], \left[\begin{array}{l} \text{qeq} \\ \text{HARG } \boxed{8} \\ \text{LARG } \boxed{9} \end{array} \right] > \end{array} \right]$$

For purposes of resolution, the existential quantifier in (40) is subject only to variable binding conditions and the requirement that it end up with some label as its body. Hence it follows without further stipulation that (40) is ambiguous: the value of **a_q_rel**'s BODY is either $\boxed{7}$, in which case **a_q_rel** outscopes **appear_rel**, or else $\boxed{9}$, in which case **appear_rel** outscopes **a_q_rel**. For essentially identical reasons, scope may be resolved in two distinct ways when quantified NPs are extracted, e.g., in (41):

(41) A unicorn, only a fool would look for.

Previous work on semantics in HPSG has relied heavily on the technique of 'Cooper Storage', a method for allowing a variable to stand in for a quantifier's contribution to the semantic content of a sentence, while the quantifier which binds that variable is placed in a 'store' (see Cooper, 1983). In the analysis presented by Pollard and Sag (1994), stored quantifiers are passed up to successively higher levels of structure until an appropriate scope assignment locus is reached (e.g., a VP or S). There, quantifier(s) may be retrieved from storage and integrated into the meaning, receiving a wide scope interpretation. This analysis entails the introduction of various features, e.g., (Q)STORE, whose value is a set of generalized quantifiers. Further constraints must be added to ensure that the set of retrieved quantifiers of any given sign constitutes a list of quantifiers that is prepended to the head daughter's QUANTS list. The data structures in Pollard and Sag's quantifier storage analysis are thus considerably more complicated than those available within an MRS analysis. This gain in descriptive simplicity, we believe, constitutes a significant advantage for MRS-based semantic analysis within HPSG.

The interaction of 'raising' and quantifier scope provides a further argument in support of an MRS analysis. Pollard and Sag's Cooper Storage analysis mistakenly assigns only one possible scope to a sentence where

a raised argument is a quantifier semantically, e.g., a sentence like (39). Because quantifier storage only allows a quantifier to obtain wider scope than the one determined by its syntactic position, there is no way to let *appear* outscope *a unicorn* in Pollard and Sag's analysis.

This problem is also solved by Pollard and Yoo (1998), but at a cost. They propose to make QSTORE a feature within *local* objects. Since the local information of a raised argument is identified with the unexpressed subject (the SUBJ element) of the raising verb's VP complement, it follows on their analysis that the quantifier stored by the NP *a unicorn* is also stored by the (unexpressed) subject of the verb *approaching* in (41). In the Pollard and Yoo analysis, the QSTORE value of each verb is the amalgamation of the QSTORE values of its arguments. Hence the existential quantifier in (41) 'starts out' in the QSTORE of *approaching* and may be retrieved inside the scope of *appear*. For example the quantifier may be retrieved into the content of the phrase *to be approaching*.

We must ensure that a quantifier never enters into storage twice, e.g., both in the QSTORE amalgamation of *approaching* and in that of *appears* in (41), as it would lead to a single quantifier entering into the meaning of the sentence twice. The cost of the Pollard-Yoo analysis is the complexity that must be introduced into the analysis in order to block this double interpretation of raised quantifiers. For example, Pollard and Yoo (1998: 421ff.) appeal to a notion of 'selected argument' which is characterized disjunctively in terms of thematic arguments selected via COMPS or SUBJ, arguments selected by SPR, or arguments selected by MOD. They also make use of the notion 'semantically vacuous', which characterizes an element whose content is shared with that of one of its complements. These notions play a role in a complex condition relating the values of the features RETRIEVED, QUANTIFIERS and POOL. The effect of this condition is to guarantee that the QSTORE value of a semantically non-vacuous lexical head is the union of the QSTORE values of all arguments that are subcategorized for and assigned a semantic role by that head. In other, more explicit formulations of this proposal, e.g., that of Przepiórkowski (1999: Chapter 9), a host of complex relations and features is introduced. Przepiórkowski's system employs the relations **selected-qs-union**, **get-selected-qs**, **selected-arg**, **thematic**, and **raised**.

This cumbersome proliferation of analytic devices is avoided entirely in the MRS analysis described above. The absence of any double interpretation of quantifiers follows directly from the facts that (1) the lexical entry for a quantificational determiner contains exactly one generalized quantifier on its RELS list and (2) the RELS list of a phrase is always the sum of the RELS list of its daughters (together with its C-CONT). Again, MRS allows the analysis of quantification to be more streamlined and more predictive than previous treatments of quantifier scope developed within HPSG.

6.5. CONTROL PREDICATES

One further HOOK feature used in the ERG is XARG (EXTERNAL-ARGUMENT). Following Sag and Pollard (1991), this feature picks out (the index of) the subject argument within a given phrase. This corresponds to the unexpressed subject in the various control environments illustrated in examples like the following:

- (42) a. The dog tried *to bark*.
 b. We tried *ignoring them*.
 c. *To remain silent* was impossible.
 d. They did it (in order) *to impress us*.
 e. They came home *quite drunk/in need of help*.
 f. *Quite drunk/In need of help*, they came home early.

The value of the feature HOOK is a feature structure that contains the semantic properties that a given phrase ‘makes visible’ to the outside. Hence, it is the information in HOOK that is available for coindexation with some external element in virtue of lexical, semantic or constructional constraints. For this reason, we locate the feature XARG within HOOK, on a par with LTOP and INDEX.

For example, the lexical entry for the verb *try* ensures that in the MRS for a sentence like *The dog tried to bark*, the ARG1 of **_try_rel** (the dog) is identified with the external argument of the infinitival complement, which is identified with the XARG value of the VP *bark*, which in turn is coindexed with the ARG1 of **_bark_rel**. This is illustrated in Figure 7.

Given the particular analysis of English control and raising embodied in the ERG, a controlled complement always has the *synsem* of the unexpressed subject on its SUBJ list. Hence, the controlled element could have been identified without XARG, by a constraint stated in terms of the following path:

LOC|CAT|SUBJ|FIRST|LOC|CONT|HOOK|INDEX

However, there are many languages (e.g., Serbo-Croatian, Halkomelem Salish) where the controlled element is realized as a pronoun within the controlled complement, whose SUBJ list is presumably empty. Under standard HPSG assumptions, such a pronoun is not accessible from outside the controlled complement. The feature XARG, by allowing the subject’s index to be visible from outside the complement, thus allows a uniform account of control in all these languages.

6.6. CONSTRUCTIONS

As we mentioned briefly above, MRS allows for the possibility that constructions may introduce EPS into the semantics. Constructions all have a

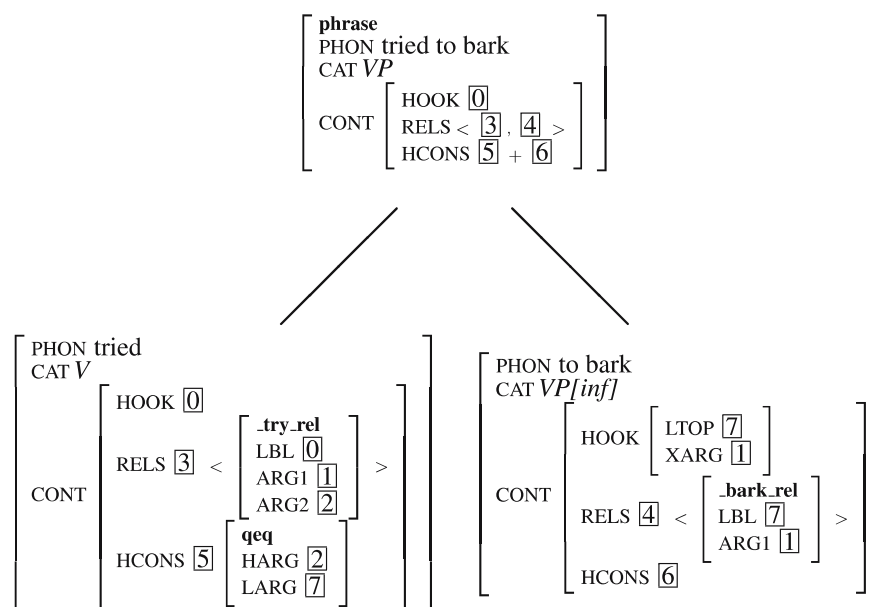


Figure 7. Control in the VP *tried to bark*.

feature **C-CONT** which takes an MRS structure as its value and behaves according to the composition rules we have previously outlined. Most constructions in the LinGO grammar actually have an empty **C-CONT**, but as an example of constructional content, we will consider the rule which is used for bare noun phrases, that is, bare plurals and bare mass nouns such as *squirrels* and *rice* in *squirrels eat rice*. The meaning of these phrases is a very complex issue (see, for example, Carlson and Pelletier, 1995) and we do not propose to shed any light on the subject here, since we will simply assume that a structure that is syntactically the same as a generalized quantifier is involved, specifically **udef_rel**, but say nothing about what it means.¹⁹ The reason for making the semantics part of the construction is to allow for modifiers: e.g., in *fake guns are not dangerous*, the rule applies to *fake guns*. The construction is shown in Figure 8 and an example of its application is in Figure 9. Notice how similar this is to the regular combination of a quantifier.

Other cases where the LinGO grammar uses construction semantics include the rules for imperative (another unary rule) and for compound nouns (a recursive binary rule). The same principles also apply for the semantic contribution of lexical rules. Eventually, we hope to be able to extend this approach to more complex constructions, such as ‘the Xer the Yer’ (e.g., *the bigger the better*), as discussed by Fillmore et al. (1988), among others.

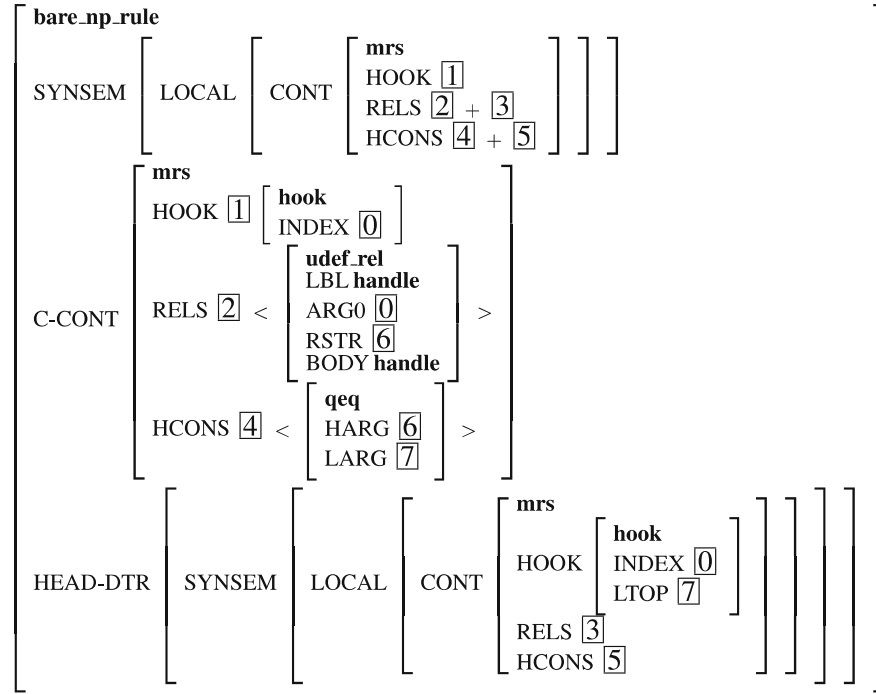
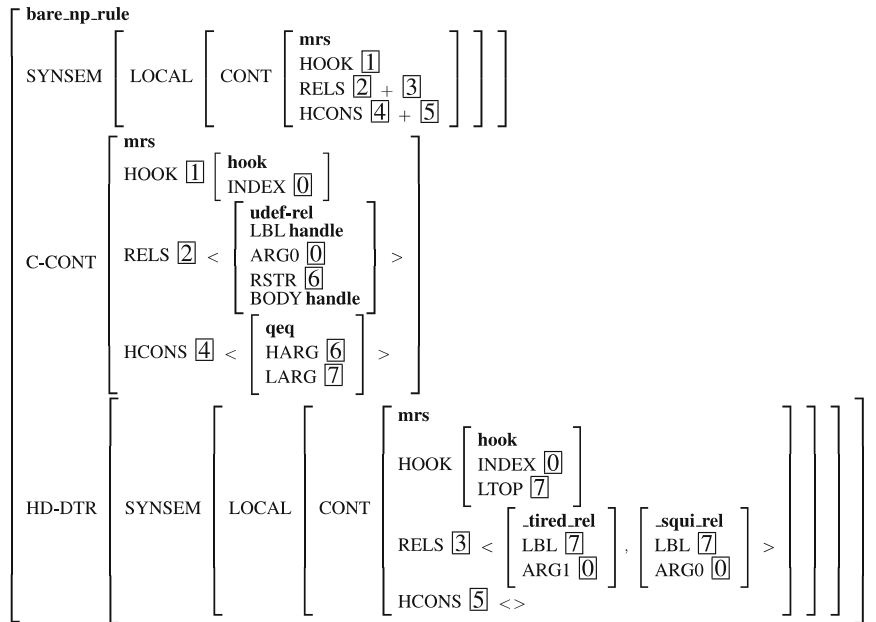
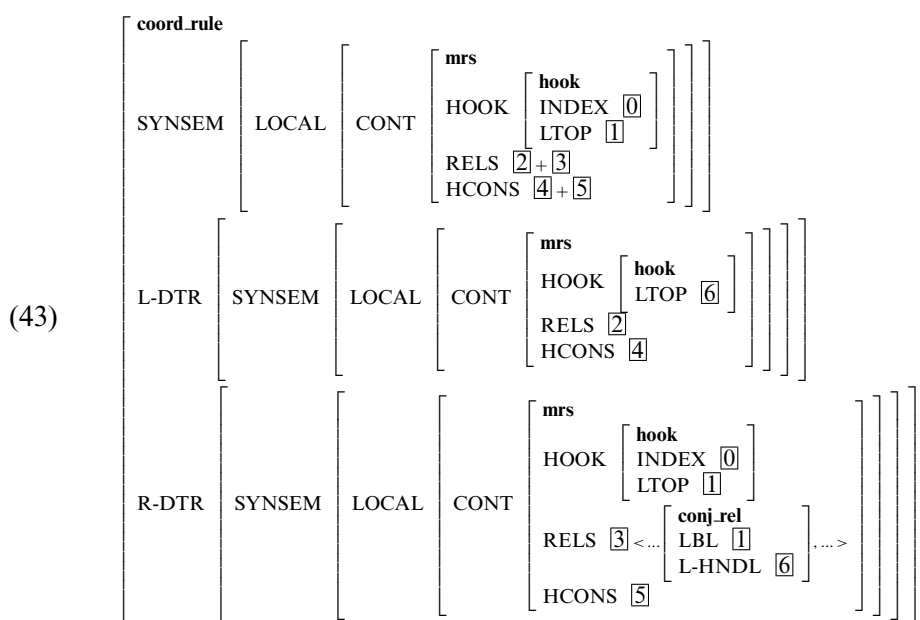


Figure 8. The bare-np rule.

Figure 9. The bare-np rule applied to *tired squirrels*.

Our analysis of coordination is based on the idea that conjunctions can be treated as binary relations, each introducing handle-valued features L-HNDL and R-HNDL for the left and right conjuncts. Our binary branching syntactic analysis adds one syntactic argument at a time, as sketched in (1).²⁰ The right conjunct is either a phrase like *and slithers* introducing a lexical conjunction, or a partial conjunction phrase like *slides and slithers* which will be part of (at least) a ternary conjunction like *slips, slides, and slithers*. These partial conjunction phrases are built with a variant of (1) in which the C-CONT introduces a new conjunction relation, keeping the semantics of conjunctions binary and fully recursive without requiring set-valued attributes.



Note that the ltop of the left conjunct is identified with the L-HNDL (left handle) argument of the conjunction relation (e.g., `_and_c_rel`, `_or_c_rel`), which is introduced onto the RELS list by the conjunct daughter on the right in any coordinate structure. Hence that right conjunct can still locally constrain the hook of the left conjunct, like any semantic head daughter does. (The ltop of the right sister to the conjunction is identified with the R-HNDL of the conj_rel when the conjunction combines with the right conjunct via a head-marker rule not shown here.) Thus our analysis yields MRS structures like the following for a ternary VP like *slip, slide, and slither*:

$$(44) \left[\begin{array}{l} \text{mrs} \\ \text{HOOK|LTOP } \boxed{1} \text{ handle} \\ \text{RELS } < \left[\begin{array}{l} \text{and_rel} \\ \text{LBL } \boxed{1} \\ \text{L-HNDL } \boxed{16} \\ \text{R-HNDL } \boxed{17} \end{array} \right], \left[\begin{array}{l} \text{slip_rel} \\ \text{LBL } \boxed{16} \\ \text{ARG1 } \boxed{9} \end{array} \right], \left[\begin{array}{l} \text{conj_rel} \\ \text{LBL } \boxed{17} \\ \text{L-HNDL } \boxed{18} \\ \text{R-HNDL } \boxed{19} \end{array} \right], \left[\begin{array}{l} \text{slide_rel} \\ \text{LBL } \boxed{18} \\ \text{ARG1 } \boxed{9} \end{array} \right], \left[\begin{array}{l} \text{slither_rel} \\ \text{LBL } \boxed{19} \\ \text{ARG1 } \boxed{9} \end{array} \right] > \\ \text{HCONS } < > \end{array} \right]$$

Identifying the ltop of each conjunct directly with the hole of one of the conj-rels (rather than via a qeq relation) enables us to avoid spurious quantifier scope ambiguity, predicting exactly the two readings available for each of the following examples (assuming a quantificational analysis for indefinites):

- (45) a. Everyone read a book and left.
b. Most people slept or saw a movie.

Since adverbs are bounded by their ltops, as we have already seen, our analysis of coordination makes the further prediction that an adverb that is clearly within a (VP, S, etc.) conjunct can never outscope the relevant **conj_rel**. As the following unambiguous examples indicate, this prediction is also correct:²¹

- (46) a. Sandy stayed and probably fell asleep.
(≠ Sandy probably stayed and fell asleep.)
c. Pat either suddenly left or neglected to call.
(≠ Pat suddenly either left or neglected to call.)

Finally, examples like the following are correctly predicted to be ambiguous, because there is a syntactic ambiguity:

- (47) He usually leaves quickly and comes in quietly.

Here the scopal adverb can modify either the left conjunct or the coordinate structure, but there is only one resolved MRS for each syntactic analysis.

7. Related Work

The idea of expressing meaning using a list of elements which correspond to individual morphemes has seemed attractive for quite some time; it is evident, for example, in Kay (1970) (though not explicitly discussed in these terms). This concept has been extensively explored by Hobbs (e.g., 1983, 1985) who developed an “ontologically promiscuous” semantics which supports underspecification of quantifier scope. In order to do this, however, Hobbs had to develop a novel treatment of quantifiers which relies on reifying the notion of the typical element of a set. Hobbs’ work is ingenious, but the complexity of the relationship with more conventional approaches makes the representations (which are often incompatible with

previous, well-worked out analyses) somewhat difficult to assess. As we discussed in §2, a form of flat semantics has been explicitly proposed for MT by Phillips (1993) and also by Trujillo (1995). However, as with Kay's work, these representations do not allow for the representation of scope.

As we described, the use of handles for representing scope in MRS straightforwardly leads to a technique for representing underspecification of scope relationships. The use of an underspecified intermediate representation in computational linguistics dates back at least to the LUNAR project (Woods et al., 1972), but more recent work has been directed at developing an adequate semantics for underspecified representations, and in allowing disambiguation to be formalized as a monotonic accumulation of constraints (Alshawi and Crouch, 1992). Of particular note in this regard is Nerbonne (1993), which describes an approach to underspecified semantic representation based on feature structures.

Despite our description of MRS in terms of predicate calculus (with generalized quantifiers), MRS is quite close to UDRT (Reyle, 1993), since handles play a role similar to that of UDRT labels. However, MRS as described here makes use of a style of scope constraint that is different from UDRT. Our motivation was two-fold: convenience of representation of the sort of constraints on scope that arise from syntax, and simplicity in semantic composition using unification. Partly because of the different notion of scope constraints, the approach to constructing UDRSs in HPSG given in Frank and Reyle (1994) is significantly different from that used here. We believe that the MRS approach has the advantage of being more compatible with earlier approaches to semantics within HPSG, making it easier to incorporate treatments from existing work.

Bos' Hole semantics (Bos, 1995) is closely related to MRS, and although the two approaches were originally developed independently, the current formalisation of MRS owes a lot to his work. From our perspective, perhaps the main difference is that MRS is strongly influenced by the demands of semantic composition with a tight syntax-semantics interface, and that MRS is explicitly designed to be compatible with a typed feature structure formalism. An alternative underspecified representation for use with typed feature structures known (a bit misleadingly) as Underspecified MRS (UMRS) was developed for a German grammar, and is described by Egg and Lebeth (1995, 1996) and Egg (1998). Abb et al., (1996) discuss the use of UMRS in semantic transfer. The Constraint-Language for Lambda Structures (CLLS) approach (Egg et al., 2001) is further from MRS, but the relationship between Hole semantics, MRS and CLLS has been elucidated by Koller et al. (2003) and Niehren and Thater (2003). Their work has also demonstrated good complexity results for satisfiability of constraints expressed in these languages. Richter and Sailer (2001) have developed an alternative to MRS for HPSG.

There is further work incorporating similar ideas into other frameworks. Kallmeyer and Joshi (1999) incorporate a semantics similar to MRS into Lexicalized Tree-Adjoining Grammar (LTAG) and Gardent and Kallmeyer (2003) demonstrate an approach to semantic construction for Feature-Based Tree Adjoining Grammar (FTAG) that is quite close to MRS semantic construction. Baldridge and Kruijff (2002) meld key ideas of MRS with Hybrid Logic²² and incorporate it into Combinatory Categorical Grammar (CCG). Dahllöf's (2002) Token Dependency Semantics builds directly on MRS.

MRS itself has been used in computational grammars other than the ERG: in particular, Siegel and Bender (2002) describe a broad coverage Japanese grammar written using MRS. Beermann and Hellan (2004) discuss the use of MRS in a Norwegian grammar which utilises more semantic decomposition than we have assumed here. Villavicencio (2002) uses MRS in a categorial grammar for experiments on child language learning. MRS is useful for this purpose because it is relatively independent of syntax and thus can be used to express possible inputs to the learner without biasing it in favor of a particular subcategorization pattern.

There is also work that indicates that flat semantics may have advantages in theoretical linguistic research with little or no computational component. Dahllöf's (2002) work formalises Davidson's paratactic approach to intensional contexts. Riehemann (2001) describes a treatment of idioms which crucially utilizes MRS. This approach treats idioms as phrasal constructions with a specified semantic relationship between the idiomatic words involved. The flatness and underspecifiability of MRS is important for similar reasons to those discussed with relation to semantic transfer in §2: some idioms appear in a wide range of syntactic contexts and the relationships between parts of the idiom must be represented in a way that is insensitive to irrelevant syntactic distinctions. Bouma et al. (1998) show that MRS semantics is a crucial ingredient of a constraint-based version of the 'adjuncts-as-complements' analysis that has been popular within HPSG.²³ They also explore an MRS analysis of classic examples like *The Sheriff jailed Robin Hood for seven years* where verbs like *jail* are lexically decomposed in terms of the two elementary predications $h1: \text{cause_become}(x, y, h2)$ and $h3: \text{in_jail}(y)$. The role that MRS plays in this work is to provide easy access to scope points in the semantics that do not correspond to any constituent within a syntactic structure. Warner (2000) shows how MRS can facilitate an account of lexical idiosyncrasy in the English auxiliary system, most notably an account of lexically idiosyncratic modal/negation scope assignments. Koenig and Muansuwan (2005) give an account of Thai aspect which also relies on MRS and Kiss (2005) uses MRS in his account of relative clause extraposition.

8. Conclusion

In our introduction, we outlined four criteria of adequacy for computational semantics: expressive adequacy, grammatical compatibility, computational tractability, and underspecifiability. In this paper, we have described the basics of the framework of Minimal Recursion Semantics. We have discussed how underspecifiability and tractability in the form of a flat representation go together, and illustrated that MRS is nevertheless compatible with conventional semantic representations, thus allowing for expressive adequacy. We have also seen how MRS can be integrated into a grammatical representation using typed feature structures.

The key ideas we have presented are the following:

1. An MRS system provides flat semantic representations that embody familiar notions of scope without explicit representational embedding.
2. Quantifier scope can be underspecified but one can specify in a precise way exactly what set of fully determinate (e.g., scoped) expressions are subsumed by any single MRS representation.
3. An HPSG can be smoothly integrated with an MRS semantics, allowing lexical and construction-specific constraints on partial semantic interpretations to be characterized in a fully declarative manner.

Because the ERG is a broad-coverage grammar, we have to provide some sort of semantic analysis for every phenomenon that arises with sufficiently high frequency in our corpora. There are many areas in which our analyses are currently inadequate and on which we intend to work further. MRS is being used as part of the Matrix, a grammar core which is intended to act as a basis on which grammars of various languages can be built (Bender et al., 2002, Bender and Flickinger, 2003). A variant of MRS, Robust MRS (RMRS) is currently under development. RMRS allows more extreme underspecification and is designed to allow shallow analysis systems to produce a highly underspecified semantic representation which is nevertheless compatible with the output of deep grammars, such as the ERG.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grants IRI-9612682 and BCS-0094638, by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant FKZ:01iV401, and by a research grant to CSLI from NTT Corporation. The final version of this paper was prepared while Sag was a Fellow at the Center for Advanced Study in the Behavioral Sciences, supported by a grant (# 2000-5633) to CASBS from The William and Flora Hewlett Foundation. Preparation of the final version was also partly supported by the DeepThought project,

funded under the Thematic Programme User-friendly Information Society of the 5th Framework Programme of the European Community (Contract No IST-2001-37836).

We are grateful for the comments and suggestions of many colleagues. In particular, Rob Malouf was responsible for the first version of MRS in the ERG, Alex Lascarides has read and commented on several drafts of the paper and helped develop the formalization of MRS composition, and Harry Bunt also influenced the way that MRS developed. We are also grateful to Emily Bender, Johan Bos, Ted Briscoe, Dick Crouch, Markus Egg, Bob Kasper, Andreas Kathol, Ali Knott, JP Koenig, Alex Koller, Stefan Müller, John Nerbonne, Joachim Niehren, Stephan Oepen, Simone Teufel, Stefan Thater and Arturo Trujillo. We are also grateful for the comments of the anonymous referees. However, as usual, the responsibility for the contents of this study lies with the authors alone.

Notes

¹ MRS was first presented in October 1994 and a partial description of it was published in Copestake et al. (1995). The current paper is closely based on a 1999 draft distributed via the authors' web pages. Several developments have occurred since then, but we have not attempted to cover them fully here, since we believe that this paper is self-contained as an introduction to MRS. §7 gives a brief overview of some work using MRS.

² This should not be taken as suggesting that we think it is impossible or undesirable to give MRS a model-theoretic semantics directly. But given that our main interest is in the computational and compositional use of MRS, it is more appropriate to show how it relates to a relatively well-known language, such as predicate calculus, rather than to attempt a direct interpretation.

³ Here and below, we use a syntax for generalized quantifiers without lambda expressions, where the first argument position is used for the bound variable. This sort of notation is common in computational semantics: see, for instance, Nerbonne (1993).

⁴ Here and below, the notion of tree we are assuming is a single-rooted connected directed graph where no node has more than one parent. This is formally equivalent to the definition in Partee et al. (1993: 16.3), though we have more complex structures than atomic symbols labeling tree nodes, and we are only interested in precedence of the daughters of a single node, rather than over the entire tree.

⁵ One place where repeated elements might arise is in a simple analysis of *big big dog*, or *yes, yes* for instance. We are not making any claims here about the desirability or otherwise of such analyses, but we do not want them to be excluded in principle by the formalism. Checking for repeated elements would increase the computational complexity of the operations on the EPS, with no useful increase in expressive power. However, we do not see this as a major issue with respect to the formalism.

⁶ If an MRS produced by a grammar for a complete sentence does not correspond to any object language structures, we assume there is an error in the grammar. There is an alternative approach which uses semantic ill-formedness to deliberately rule out analyses, but this requires a rather unlikely degree of perfection on the part of the grammar developer in order to be really practical.

⁷ In §4.2 we will introduce a form of constraint that is slightly more complex than a simple outscope constraint.

⁸ It is possible to define certain non-monotonic forms of MRS, but we will not consider that further in this paper.

⁹ Indeed, in earlier versions of MRS, we used different types of constraints from those described here.

¹⁰ We are taking the variable binding condition into account here: if we do not, there are even more possible scopes.

¹¹ It turns out that the conditions on handles in semantic composition described in the next section mean that for all the examples we will look at in this paper, it is unnecessary to use qeq relationships rather than simple outscopes. These conditions were developed some time after we introduced the concept of qeq and may in the long term make it possible to use outscopes conditions instead. We hypothesise that in a grammar which perfectly obeyed the constraints on composition, it should be unnecessary to use qeq conditions rather than simple outscopes. However, in actual grammars we find that there is a difference, since the implemented grammars sometimes do not completely meet the constraints described. Furthermore, to our knowledge, it has not yet been proven that in a grammar where these conditions on composition were perfectly obeyed, qeq constraints would be equivalent to outscopes. We therefore continue to use qeq conditions. See also Fuchss et al., (2004).

¹² For simplicity in this section, we show the lexical entries as having empty handle constraints, but in a lexicalist grammar partially specified handle constraints may be present in the lexical entries.

¹³ For discussion, see Hobbs and Shieber (1987)

¹⁴ In earlier versions of MRS, the two features that we have called G_{TOP} and R_{ELS} were called H_{ANDEL} (or H_{NDL}) and L_{ISZT} (or L_{ZT}). L_{BL} was also called H_{ANDEL} or H_{NDL}.

¹⁵ Called a ‘hole’ by Copestake et al. (2001).

¹⁶ See also: <http://lingo.stanford.edu/>.

¹⁷ The attribute and type names used in this paper reflect the version of the ERG as of July 2003. Since then we have introduced further normalisation of predicate names, which is important from the perspective of maintaining the grammar and interfacing it with other systems. However, detailed discussion of this would take us too far from the current paper.

¹⁸ The global top (or conceivably even a set of non-local tops) may however play an important role in DRT-style semantic representations (Kamp and Reyle, 1993: p. 107 and chapter 3), for example enabling a constraint that all pronoun or proper name handles be identified with the global top handle.

¹⁹ In particular, we will offer no account of the distinction between ‘weak’ and ‘strong’ generic sentences. We would argue, however, that the meaning of such phrases is context-dependent. For instance, *squirrels eat roses* is clearly a generic sentence, but at least when uttered in response to *What on earth happened to those flowers?*, it does not have to mean that most squirrels are rose eaters. On the other hand, *dogs are herbivores* seems clearly false, even though there are some dogs on a vegetarian diet. It is therefore appropriate, given our general approach, to be as neutral as possible about the meaning of such phrases in the grammar. But we won’t enter into a discussion about whether it is appropriate to use a true generalized quantifier, as opposed to some variety of kind reading, for instance, since the point here is simply the mechanics of phrasal contributions to the semantics.

²⁰ This presentation simplifies the implemented ERG coordination analysis in non-essential respects.

²¹ However, we have not presented an account here of the apparent quantifier ‘duplication’ in examples like *We needed and craved more protein*.

²² For more on Hybrid Logic, see: <http://www.hylo.net>.

²³ For an overview of such analyses see, Przepiórkowski, (1999).

References

- Abb, B., Buschbeck-Wolf, B. and Tschernitschek, C. (1996) Abstraction and Underspecification in Semantic Transfer. In: *Proceedings of the Second Conference of the Association for Machine Translation in the Americas (AMTA-96)*. Montreal, pp. 56–65.
- Alshawi, H. and Crouch, R. (1992) Monotonic Semantic Interpretation. In: *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-92)*. Newark, NJ, pp. 32–39.
- Baldrige, J. and Kruijff, G.-J. M. (2002) Coupling CCG and Hybrid Logic Dependency Semantics. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia.
- Beermann, D. and Hellan, L. (2004) Semantic Decomposition in a Computational HPSG Grammar: A Treatment of Aspect and Context-dependent Directionals. In: *Proceedings of the Workshop on Computational Semantics at the 11th International Conference on HPSG* Leuven, Belgium.
- Bender, E. M., Flickinger, D. and Oepen, S. (2002) The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In: *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*. Taipei, Taiwan, pp. 8–14.
- Bender, E. M. and Flickinger, D. (2003) Compositional Semantics in a Multilingual Grammar Resource. In: *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development, ESSLLI 2003*. Vienna, Austria.
- Bos, J. (1995). Predicate Logic Unplugged. In: *Proceedings of the 10th Amsterdam Colloquium*. Amsterdam, pp. 133–142.
- Bouma, G., Malouf, R. and Sag, I. A. (1998) Adjunct Scope and Complex Predicates, Paper presented at the 20th Annual Meeting of the DGfS. Section 8: The Syntax of Adverbials – Theoretical and Cross-linguistic Aspects. Halle, Germany.
- Carlson, G. N. and Pelletier, F. J. (eds.) (1995) *The Generic Book*. University of Chicago Press, Chicago.
- Carroll, J., Copestake, A., Flickinger, D. and Poznanski, V. (1999) An Efficient Chart Generator for (Semi-)lexicalist Grammars. In: *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG'99)*. Toulouse, pp. 86–95.
- Cooper, R. (1983) *Quantification and Syntactic Theory*. Reidel, Dordrecht.
- Copestake, A. (1995) Semantic Transfer for Verbmobil. ACQUILEX II working paper 62 and Verbmobil report 93.
- Copestake, A., Flickinger, D., Malouf, R., Riehemann, S. and Sag, I. A. (1995) Translation using Minimal Recursion Semantics. In: *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*. Leuven, Belgium.
- Copestake, A., Lascarides, A. and Flickinger, D. (2001) An Algebra for Semantic Construction in Constraint-Based Grammars. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*. Toulouse, France.
- Dahllöf, M. (2002) Token Dependency Semantics and the Paratactic Analysis of Intensional constructions. *Journal of Semantics* 19, pp. 333–368.
- Davis, A. (2001) *Linking by Types in the Hierarchical Lexicon*. CSLI Publications, Stanford, CA.
- van Deemter, K. and Peters, S. (eds.) (1996) *Semantic Ambiguity and Underspecification*. CSLI Publications, Stanford, CA.
- Egg, M. (1998) *Wh*-questions in Underspecified Minimal Recursion Semantics. *Journal of Semantics* 15(1), pp. 37–82.

- Egg, M. and Lebeth, K. (1995) Semantic Underspecification and Modifier Attachment. *Intergrative Ansätze in der Computerlinguistik. Beiträge zur 5. Fachtagung für Computerlinguistik der DGfS*.
- Egg, M. and Lebeth, K. (1996) Semantic Interpretation in HPSG. Presented at the Third International Conference on HPSG. Marseilles, France.
- Egg, M., Koller, A. and Niehren, J. (2001) The Constraint Language for Lambda Structures. *Journal of Logic, Language, and Information* 10, pp. 457–485.
- Fillmore, C. J., Kay, P. and O'Connor, M. C. (1988) Regularity and Idiomatcity in Grammatical Constructions: The Case of *Let Alone*. *Language* 64(3), pp. 501–538.
- Flickinger, D., Copestake, A. and Sag, I. A. (2000) HPSG Analysis of English. In: W. Wahlster and Karger, R. (ed.), *VerbMobil: Foundations of Speech-to-Speech Translation*. Springer Verlag, Berlin, Heidelberg and New York, pp. 254–263.
- Frank, A. and Reyle, U. (1994) Principle-Based Semantics for HPSG. *Arbeitspapiere des Sonderforschungsbereichs 340*. University of Stuttgart.
- Fuchss, R., Koller, A., Niehren, J. and Thater, S. (2004) Minimal Recursion Semantics as Dominance Constraints: Translation, Evaluation, and Analysis. In: *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL-04)*. Barcelona, Spain, pp. 247–254.
- Gardent, C. and Kallmeyer, L. (2003) Semantic Construction in Feature-Based TAG. In: *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*. Budapest, pp. 123–130.
- Ginzburg, J. and Sag, I. A. (2000) *Interrogative Investigations: The Form, Meaning and Use of English Interrogatives* CSLI. Stanford.
- Hobbs, J. (1983) An Improper Treatment of Quantification in Ordinary English. In: *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics (ACL-83)*. MIT, Cambridge, MA, pp. 57–63.
- Hobbs, J. (1985) Ontological Promiscuity. In: *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL-85)*. Chicago, IL, pp. 61–69.
- Hobbs, J. and Shieber, S.M. (1987) An Algorithm for Generating Quantifier Scopings. *Computational Linguistics* 13, pp. 47–63.
- Kallmeyer, L. and Joshi, A. (1999) Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. In: *Proceedings of the 12th Amsterdam Colloquium*. Dekker, P. (ed.) Amsterdam, ILLC, pp. 169–174.
- Kamp, H. and Reyle, U. (1993) *From Discourse to Logic: An Introduction to Modeltheoretic Semantics, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Kasper, R. (1996) Semantics of Recursive Modification, unpublished MS. Ohio State University.
- Kay, M. (1970) From Semantics to Syntax. In: Bierwisch, M. and Heidorn, K. E. (eds.), *Progress in Linguistics*, The Hague, Mouton, pp. 114–126.
- Kay, M. (1996) Chart Generation. In: *Proceeding of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*. Santa Cruz, CA, pp. 200–204.
- Kiss, T. (2005) Semantic Constraints on Relative Clause Extraposition. *Natural Language and Linguistic Theory* 23(2), pp. 281–334.
- Koenig, J.-P. and Muansuwan, N. (2005) The Syntax of Aspect in Thai. *Natural Language and Linguistic Theory* 23(2), pp. 335–380.
- Koller, A., Niehren, J. and Thater, S. (2003) Bridging the Gap between Underspecification Formalisms: Hole Semantics as dominance constraints. In: *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*. Budapest, pp. 195–202.

- Landsbergen, J. (1987) Isomorphic Grammars and Their Use in the ROSETTA Translation System. In: King, M. (ed.), *Machine Translation Today: The State of the Art*. Edinburgh University Press, pp. 351–372.
- Nerbonne, J. (1993) A Feature-Based Syntax/Semantics Interface. *Annals of Mathematics and Artificial Intelligence* (Special Issue on Mathematics of Language, Manaster-Ramer, A. and Zadrozny, W. eds.), 8, pp. 107–132.
- Niehren, J. and Thater, S. (2003) Bridging the Gap Between Underspecification Formalisms: Minimal Recursion Semantics as Dominance Constraints. In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*. Sapporo, Japan.
- Partee, B., ter Meulen, A. and Wall, R. E. (1993) *Mathematical Methods in Linguistics*. Kluwer Academic Publishers, Dordrecht.
- Phillips, J. D. (1993) Generation of Text from Logical Formulae. *Machine Translation* 8(4), pp. 209–235.
- Pinkal, M. (1996) Radical Underspecification, In: Dekker, P. and Stokhof, M. (eds.), *Proceedings of the 10th Amsterdam Colloquium*. Amsterdam, ILLC, pp. 587–606.
- Pollard, C. and Sag, I. A. (1994) *Head-driven Phrase Structure Grammar*. University of Chicago Press, Chicago.
- Pollard, C. and Yoo, E. J. (1998) A Unified Theory of Scope for Quantifiers and *Wh*-phrases. *Journal of Linguistics* 34(2), pp. 415–445.
- Poznanski, V., Beaven, J. L. and Whitelock, P. (1995) An Efficient Generation Algorithm for Lexicalist MT. In: *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*. Cambridge, MA, pp. 261–267.
- Przepiórkowski, A. (1999) *Case Assignment and the Complement-Adjunct Dichotomy – A non-configurational, constraint-based approach*. Doctoral dissertation, Universität Tübingen.
- Reyle, U. (1993) Dealing with Ambiguities by Underspecification. *Journal of Semantics* 10, pp. 123–179.
- Richter, F. and Sailer, M. Polish Negation and Lexical Resource Semantics. In: Kruijff, G.-J. M., Moss, L. S. and Oehle, R. T. (eds.), *Proceedings of FGMOL 2001. Electronic Notes in Theoretical Computer Science, Vol. 53*. Elsevier.
- Riehmann, S. (2001) *A Constructional Approach to Idioms and Word Formation*. Doctoral dissertation, Stanford University.
- Sag, I. A. and Pollard, C. (1991) An Integrated Theory of Complement Control. *Language* 67(1), pp. 63–113.
- Shieber, S. M. (1993) The Problem of Logical form Equivalence. *Computational Linguistics* 19(1), pp. 179–190.
- Siegel, M. and Bender, E. M. (2002) Efficient Deep Processing of Japanese. In: *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization. COLING 2002 Post-Conference Workshop*. Taipei, Taiwan.
- Trujillo, I. A. (1995) *Lexicalist Machine Translation of Spatial Prepositions*. PhD dissertation, University of Cambridge.
- Villavicencio, A. (2002) *The Acquisition of a Unification-Based Generalised Categorical Grammar*. PhD dissertation, University of Cambridge, available as Computer Laboratory Technical Report 533.
- Warner, A. (2000) English Auxiliaries without Lexical Rules. In: Borsley, R. (ed.), *Syntax and Semantics, Vol. 32: The Nature and Function of Syntactic Categories*, Academic Press, San Diego and London, pp. 167–220.
- Whitelock, P. (1992) Shake-and-Bake Translation. In: *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*. Nantes, France.

Woods, W., Kaplan, R. M. and Nash-Webber, B. (1972) *The LUNAR Sciences Natural Language Information System: Final Report* (BBN Technical Report 2378). Bolt, Beranek and Newman Inc., Cambridge, MA.