Minimal Recursion Semantics
An Introduction

DRAFT of September 1999

Ann Copestake, Dan Flickinger, Ivan A. Sag
CSLI, Stanford University
Stanford, CA 94305, USA
{aac,dan,sag}@csli.stanford.edu
Carl Pollard

Abstract

We describe minimal recursion semantics (MRS), a framework for computational semantics which simplifies the design of algorithms. We have integrated an implementation of MRS with an HPSG grammar for both parsing and generation that enables a simple formulation of the grammatical constraints on lexical and phrasal semantics. 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.

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 (ways of 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 recent work on developing representations which allow scope to be underspecified. See, for instance, Alshawi and Crouch (1992), Reyle (1993), Pinkal (1996) and the papers collected in van Deemter and Peters (1996) and in Crouch and Poesio (1996) (though we will not assume familiarity with 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 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 more purely theoretical linguistic perspective (e.g., Riehemann (1996), Sag (1998), Warner (1999)).

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. §7 discusses related work and in §8 we conclude by discussing some outstanding issues.

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 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 just give a very 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

[1] MRS was first presented in October 1994, and a partial description of it was published in Copestake et al (1995). Other work using MRS is listed in §7.

[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 since this paper is intended as a relatively non-technical introduction to 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.

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 $[\text{white}(x) \wedge (\text{English}(x) \wedge \text{horse}(x))]$

Both of these examples simply involve conjunction, but there are much more complex cases. For instance, the English phrase *beginning of spring* can be translated in to 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
def(x, spring(x), the(y, beginning(y,x), arrive(y)))
the(y, def(x, spring(x), beginning(y,x)), 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 most 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 practicable 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 proved 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 (e.g., Whitelock, 1992). Shake-and-Bake operates by considering transfer between bags of instantiated lexical signs rather than LFs, which avoids the sort of problems that 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 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 later 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
$\text{fierce}(x), \text{black}(x), \text{cat}(x)$
b the beginning of spring arrives
$\text{the}(y), \text{beginning}(y, x), \text{def}(x), \text{spring}(x), \text{arrive}(e, y)$

Note that the structure is a list of elementary predications, which can be taken as being conjoined. Adverbs are all assumed to take events as arguments. 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.

[3] Here and below, we use a fairly standard syntax for generalized quantifiers, where the first argument position is used for the bound variable.

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 (7) it is impossible to retrieve the correct reading unambiguously, because we would get the same structure for (8), for instance.

(7) $\text{every}(x), \text{horse}(x), \text{old}(x), \text{white}(x)$

(8) 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 which 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 which is sufficiently powerful to represent natural language, it seems that this may not be the most practically 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 which 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 which 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 multiple authors (e.g. Hobbs 1983).

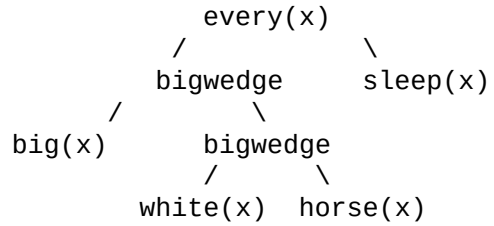
Before we go into detail about how MRS fulfills these requirements, we should emphasize that, although we have concentrated on motivating flat semantics by considering semantic transfer, we believe that the lessons learned are relevant for multiple applications in NLP. Shieber (1993) discusses logical 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 essentially 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 which 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, (9).

(9) a every big white horse sleeps

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

c


This 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 write $\bigwedge(\text{white}(x), \text{horse}(x))$ rather than $\text{white}(x) \wedge \text{horse}(x)$.

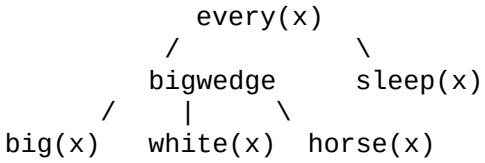
2. All non-scopal arguments in a predication precede all scopal arguments.

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]

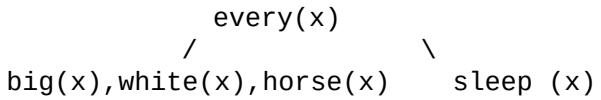
As we mentioned above, we assume a three argument notation for generalized quantifiers. We will use the terminology *restriction* and *body* to refer to the two set arguments, where we use *body* rather than *scope* to avoid confusion between this and the more general concept of scope.

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 (9a), rather than (9b)/(9c), we can use (10a)/(10b).

(10) a $\text{every}(x, \bigwedge(\text{big}(x), \text{white}(x), \text{horse}(x)), \text{sleep}(x))$

b


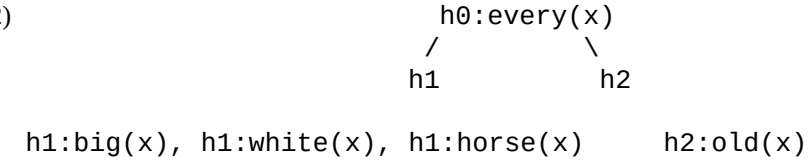
Our next move is to simplify this notation by omitting the conjunction symbol, on the basis of a convention whereby a group of elementary predications on a tree node is always assumed to be conjoined. Thus the equivalent of the expression above is shown in (11).

(11)


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 should be 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, by using tags which match up scopal argument slots with the EPs (or conjunctions of EPs) that fill them. We refer to these tags as *handle*s, 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 slots. 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 (12).

(12)


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 (12) 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 (13).

[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 actually interested in precedence of the daughters of a single node, rather than over the entire tree.

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

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. The structure in (14) is equivalent to the one given in (13).

(14) $h1\colon \text{white}(x), h0\colon \text{every}(x, h1, h2), h1\colon \text{big}(x), h2\colon \text{old}(x), h1\colon \text{horse}(x)$

The choice of particular names for handles is also semantically irrelevant, so we could equally well write (15) (which is an alphabetic variant of (14)).

(15) $h0\colon \text{white}(x), h1\colon \text{every}(x, h0, h3), h0\colon \text{big}(x), h3\colon \text{old}(x), h0\colon \text{horse}(x)$

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, it is just a syntactic variant of the notation we started with, since if we want to draw a conventional tree we just join the pieces up 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, the MRS notation as described so far does not allow for predicate modifiers. For instance, there is no way of specifying something like $Kind(beer)(x)$, which might be used, for instance, in the representation of *this pub sells 25 beers* where *25 beers* means *25 kinds of beer*. We return to this issue briefly in §8 but for now we will assume that predicate modifiers are not allowed in the object language.

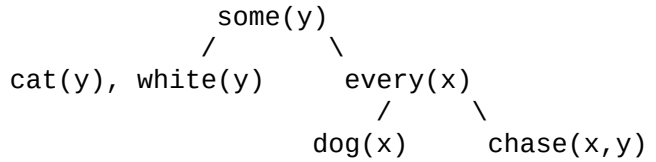
These flat labeled structures are MRSs, but the reason why MRS is not a notational variant of the conventional representation is that, rather than just using fully determined links in the tree such that handles in the argument positions are identical to labels on EPs, we can allow for underspecification of the links in order to represent multiple scopes. 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 (16).

(16) every dog chases some white cat

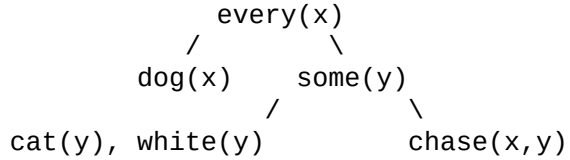
This has the fully specified readings shown in (17) (wide scope *some*) and in (18) (wide scope *every*), where in both cases we show the conventional notation, the tree (with implicit conjunction) and the MRS equivalent:

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

b 

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

(18) a $\text{every}(x, \text{dog}(x), \text{some}(y, \text{white}(y) \wedge \text{cat}(y), \text{chase}(x, y)))$

b 

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

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 which are constant between the two readings, and stipulating some constraints on how they may be joined together. The basic conditions on joining the pieces are those that are enforced by the predicate calculus representation: there must be no cases where an argument is left unsatisfied and an EP may only fill at most one argument position. Given these conditions, we can generalize over these two structures as shown in (19).

(19) a $h1$: every$(x, h3, hA), h3$: dog$(x), h7$: white$(y), h7$: cat$(y), h5$: some$(y, h7, hB), h4$: chase(x, y)

For expository purposes for this example, we have used hA and hB for the handles in the body positions of the quantifiers.

In terms of the tree representation, what we have done is replaced 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 $hA = h5$ and $hB = h4$ or so that $hA = h4$ and $hB = h5$. But we couldn't make $hA = hB = h4$, 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. [5] If the set contains more than one element, then we say that the MRS is underspecified (with respect to the object language).

So, to summarize this section, there are two basic representational tricks in 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 other trick is to recognize the special status of the logical connective 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 trick but kept an explicit conjunction relation. However implicit conjunction combined with scope reification considerably facilitates generation and transfer using MRS (see Copestake et al, 1995, and 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 which will essentially correspond 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*

This is written as handle: relation$(\text{arg}_1 \ldots \text{arg}_n, \text{scarg}_1 \ldots \text{scarg}_n)$. For instance $h2$: 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 handle-taking arguments of E is the label of E'. The *outscopes* relation is the transitive closure of the immediately outscopes

[5] If an MRS produced by a grammar for a complete sentence does not correspond to any object language structures, we assume there's 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 in the grammar developer to be really practical.

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 slots.

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. The *handle constraints* or *hcons* slot 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 be obeyed by the EPs without defining what form the constraints take.

Later on, we will see that two further slots are needed for semantic composition: the *local top* and the *index*.

Definition 3 (MRS Structure) *An MRS structure is a tuple $\langle T, L, C\rangle$ where T is a handle, L is a bag of* EPs *and C is a bag of handle constraints, such that:*

Top *There is no handle h that outscopes T.*

Handle constraints *The outscopes order between the* EPs *in L respects any constraints in C.*

For example, the MRS structure in (19) can be more precisely written as (20) where we have explicitly shown the top handel, $h0$, and the empty bag of handle constraints. 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.

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

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 both 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 are identified with an EP label.*

It follows from these conditions that in a scoped MRS structure all EP labels must be identified with a handle argument in some EP, except for the case of the top handle, which must be the label of the top node of the tree. As we have saw, 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 the 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. The following MRS is not scope-resolved, even though there is trivially only one object-language expression to which it could correspond.

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

In order to formalize the idea that an underspecified MRS will correspond to a set of more than one object-language expression, we will define the relationship of an arbitary MRS structure to a set of scope-resolved MRSs. The intuition is that the pieces of 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 MRSs. Linking simply consists of adding equalities between handels. Linking can be regarded as a form of specialization, and it is

always monotonic.[6] 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 (20) strictly link-subsumes the structure shown in (22a) and is strictly link-subsumed by the structure in (22b).

(22) a $\langle h0, \{h1\colon \text{every}(x, h2, h3), h2\colon \text{dog}(x), h3\colon \text{chase}(x, y), h5\colon \text{some}(y, h6, h7), h6\colon \text{white}(y), h6\colon \text{cat}(y)\}, \{\}\rangle$

b $\langle h0, \{h1\colon \text{every}(x, h2, h3), h2\colon \text{dog}(x), h4\colon \text{chase}(x, y), h5\colon \text{some}(y, h6, h7), h8\colon \text{white}(y), h8\colon \text{cat}(y)\}, \{\}\rangle$

Interesting linkings involve one of three possibilities:

1. Equating handle-taking argument positions in EPs with labels of other EPs.
2. Equating labels of EPs to form a larger EP conjunction.
3. Equating the top handel with the label of an EP conjunction.

It is also possible to equate two handle-taking argument positions, 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, $M_1, M_2, \ldots M_n$, 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 $O_1, O_2, \ldots O_n$. 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 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 argument slots, corresponding to the restriction and body arguments. The basic assumptions are essentially the same as in any representation with generalized quantifiers. 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, it is unbound. No two quantifiers may share bound variables. Then a scope-resolved MRS without unbound variables will correspond to a statement in predicate calculus.

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 (scoped-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 variables 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 the feature structure representation. So, since there is no loss of generality, we assume distinct variables are always represented by distinct names in an MRS, and 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 semantics of individual natural language phrases. We discuss this in the next section.

[6] There is actually some scope for defining non-monotonic forms of MRS (e.g., Copestake (1996)), but we will not consider that further in this paper.

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.[7] 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 (20) 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). This means the quantifier restriction must be underspecified in the MRS representation. However, if this is left unconstrained, as in, for instance, (23c), unwanted scopes can be obtained, specifically those in (23d), as well as the desirable scopes in (23b).[8]

(23) a every nephew of some fierce aunt runs

b $\text{every}(x, \text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{nephew}(x, y)), \text{run}(x))$
$\text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{every}(x, \text{nephew}(x, y), \text{run}(x)))$

c $\langle h1, \{h2\colon \text{every}(x, h3, h4), h5\colon \text{nephew}(x, y), h6\colon \text{some}(y, h7, h8), h7\colon \text{aunt}(y), h7\colon \text{fierce}(y), h10\colon \text{run}(x)\}, \{\}\rangle$

d $\text{every}(x, \text{run}(x), \text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{nephew}(x, y)))$
$\text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{every}(x, \text{run}(x), \text{nephew}(x, y)))$

There are a variety of forms of constraint 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 a *qeq* constraint (or $=_q$) which stands for equality modulo quantifiers. A qeq constraint always relates a handle in an argument position to a label. The intuition is that if a handle argument, h, is qeq some label, l, either that argument slot must be directly filled by l (i.e., $h = l$), or one or more quantifiers float in between h and l, such that the label of the quantifier fills the argument position and the body argument of the 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) $\langle h1, \{h2\colon \text{every}(x, h3, h4), h5\colon \text{nephew}(x, y), h6\colon \text{some}(y, h7, h8), h9\colon \text{aunt}(y), h9\colon \text{fierce}(y), h10\colon \text{run}(x)\},$
$\{h1 =_q h10, h7 =_q h9, h3 =_q h5\}\rangle$

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 some label, l, just in case $h = l$ or there is some (non-repeating) chain of one or more quantifier* EPs *E_1, $E_2 \ldots E_n$ 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
$\langle h0, \{h1\colon \text{every}(x, h2, h3), h4\colon \text{dog}(x), h5\colon \text{probably}(h6), h7\colon \text{chase}(x, y), h8\colon \text{some}(y, h9, h10), h11\colon \text{white}(y), h11\colon \text{cat}(y)\},$
$\{h0 =_q h5, h2 =_q h4, h6 =_q h7, h9 =_q h11\}\rangle$

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

[7] Indeed, in earlier versions of this work we used different types of constraints from those described here.

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

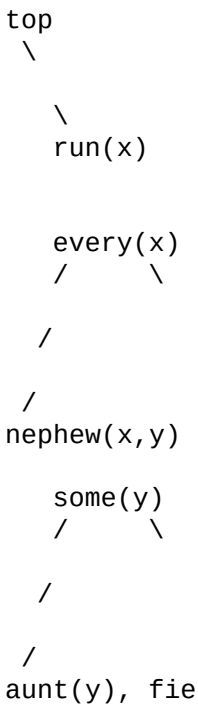


Figure 1: Schematic representation of the qeq relationships in *every nephew of some fierce aunt runs*

(26) probably(every(x, dog(x), some(y, white(y) $\wedge$ cat(y), chase(x, y))))
every(x, dog(x), probably(some(y, white(y) $\wedge$ cat(y), chase(x, y))))
every(x, dog(x), some(y, white(y) $\wedge$ cat(y), probably(chase(x, y))))
probably(some(y, white(y) $\wedge$ cat(y), every(x, dog(x), chase(x, y))))
some(y, white(y) $\wedge$ cat(y), probably(every(x, dog(x), chase(x, y))))
some(y, white(y) $\wedge$ cat(y), every(x, dog(x), probably(chase(x, y))))

Notice that, although *probably* is scopal, it cannot be inserted into a $=_q$ relationship. We can 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 succinctly capturing the constraints on scope that otherwise could require mechanisms for quantifier storage. In §8 we will raise the issue of whether other EPs besides quantifiers should be allowed to float, and whether other types of constraint are needed in addition to *qeq*.

4.3 Semantic composition

When MRS is used in a grammar, there are constraints on the relationship between the MRS of a phrase and the MRSs of its daughter(s). We will concentrate on specifying the way that the handle relationships are built up, since the details of how ordinary variables are related to each other is largely orthogonal to the use of MRS. In order to state the rules of combination, we will assume that MRSs have an extra slot, the *local top*, or *ltop*, which will be the topmost label in an MRS which is not the label of a floating EP. For instance, in (25), the ltop is the label corresponding to the EP for probably, $h5$. In a complete MRS, the top handle will be qeq 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, 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 which only contain floating EPs, the ltop is not bound to any label.

So an MRS structure is a tuple $\langle T, LT, L, C\rangle$ where LT is the local top and T, L and C are the (global) top, EP

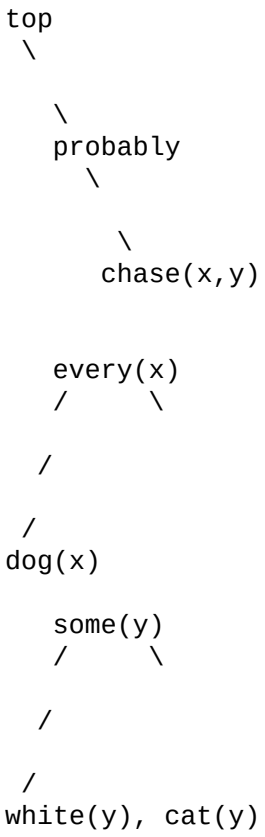


Figure 2: Schematic representation of the qeq relationships in *every dog proabably chases some white cat*

bag and handle constraints, as before. The composition rules will mostly concern the ltop rather than the global top. For convenience, in the representations in this section, we will consistently use $h0$ for the global top.

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.[9] There are three cases to consider:

1. The key is a non-scopal and fixed scopal EP. In this case, the ltop handle of the MRS is equal to the handle of the key EP. For example, $\langle h0, h1, \{h1\colon \text{dog}(x)\}, \{\}\rangle$, $\langle h0, h2, \{h2\colon \text{probably}(h3)\}, \{\}\rangle$).[10]

2. The key is a floating EP. In this case, the ltop handle of the MRS is not equated to any handle. For example, $(\langle h0, h3, \{h4\colon \text{every}(y, h5, h6)\}, \{\}\rangle)$.

3. If the lexical item has an empty EP bag its MRS structure has an unequated ltop handle: $\langle h0, h1, \{\}, \{\}\rangle$.

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 ltop handles of the daughters are equated with each other and with the ltop handle of the phrase's MRS. That is, if the MRS for the mother is $\langle T_0, LT_0, L_0, C_0\rangle$ and the daughters are $\langle T_1, LT_1, L_1, C_1\rangle \ldots \langle T_n, LT_n, L_n, C_n\rangle$, then $T_0 = T_1 = T_2 \ldots T_n$, $LT_0 = LT_1 = LT_2 \ldots LT_n$, $L_0 = L_1 + L_2 \ldots L_n$, $C_0 = C_1 + C_2 \ldots C_n$ (where + stands for append). For example if the MRS for *white* is $\langle h0, h1, \{h1\colon \text{white}(x)\}, \{\}\rangle$ and that for *cat* is $\langle h0, h2, \{h2\colon \text{cat}(y)\}, \{\}\rangle$ then the MRS for *white cat* is $\langle h0, h1, \{h1\colon \text{cat}(x), h1\colon \text{white}(x)\}, \{\}\rangle$.

2. Scopal combination. In the straightforward case, this involves a binary rule, with one daughter containing a scopal EP which scopes over the other daughter. The handle-taking argument of the scopal EP is stated to be qeq the ltop handle of the scoped-over phrase. The ltop handle of the phrase is the ltop handle of the MRS which contains the scopal EP. That is, if the MRS for the mother is $\langle T_0, LT_0, L_0, C_0\rangle$, the scoping daughter is $\langle T_s, LT_s, \{T_s\colon E(\ldots, h, \ldots)\ldots\}, C_s\rangle$ and the scoped-over daughter is $\langle T_{ns}, LT_{ns}, L_{ns}, C_{ns}\rangle$, then $T_0 = T_s = T_{ns}$, $LT_0 = LT_s$, $L_0 = L_s + L_{ns}$ and $C_0 = C_s + C_{ns} + \{h =_q LT_{ns}\}$. For example, if the MRS for *sleeps* is $\langle h0, h5, \{h5\colon \text{sleep}(x)\}, \{\}\rangle$ and for *probably* $\langle h0, h2, \{h2\colon \text{probably}(h3)\}, \{\}\rangle$ the result for *probably sleeps* would be $\langle h0, h2, \{h2\colon \text{probably}(h3), h5\colon \text{sleep}(x)\}, \{h3 =_q h5\}\rangle$. The extension for rules with multiple daughters where one scopal EP takes two or more arguments is straightforward: there will be a separate qeq 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, given *every* is $\langle h0, h1, \{h2\colon \text{every}(x, h3, h4)\}, \{\}\rangle$ and *dog* is $\langle h0, h5, \{h5\colon \text{dog}(y)\}, \{\}\rangle$ then *every dog* is $\langle h0, h1, \{h2\colon \text{every}(x, h3, h4), h5\colon \text{dog}(y)\}, h3 =_q h5\rangle$.

3. Root. The root condition stipulates that the global top is qeq the local top of the root phrase. That is if the MRS for the sentence is $\langle T_0, LT_0, L_0, C_0\rangle$ then C_0 must include $T_0 =_q LT_0$.

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 specified how this happens in the composition rules above since the details are grammar-specific.

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 allowed in MRS, since the contribution of the construction is simply treated as an extra daughter. Examples of this will be discussed in §6.5.

The effect of the semantic composition rules stated above can best be seen figuratively, using 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

[9] In the sort of grammars we write, there will usually only be one EP for a lexical item, which is therefore unambiguously the key.

[10] For simplicity in this section, we show the lexical entries as having empty handle constraint slots, but in a lexicalist grammar partially specified handle constraints may be present in the lexical entries. Since this is grammar-specific, we leave discussion to §6.

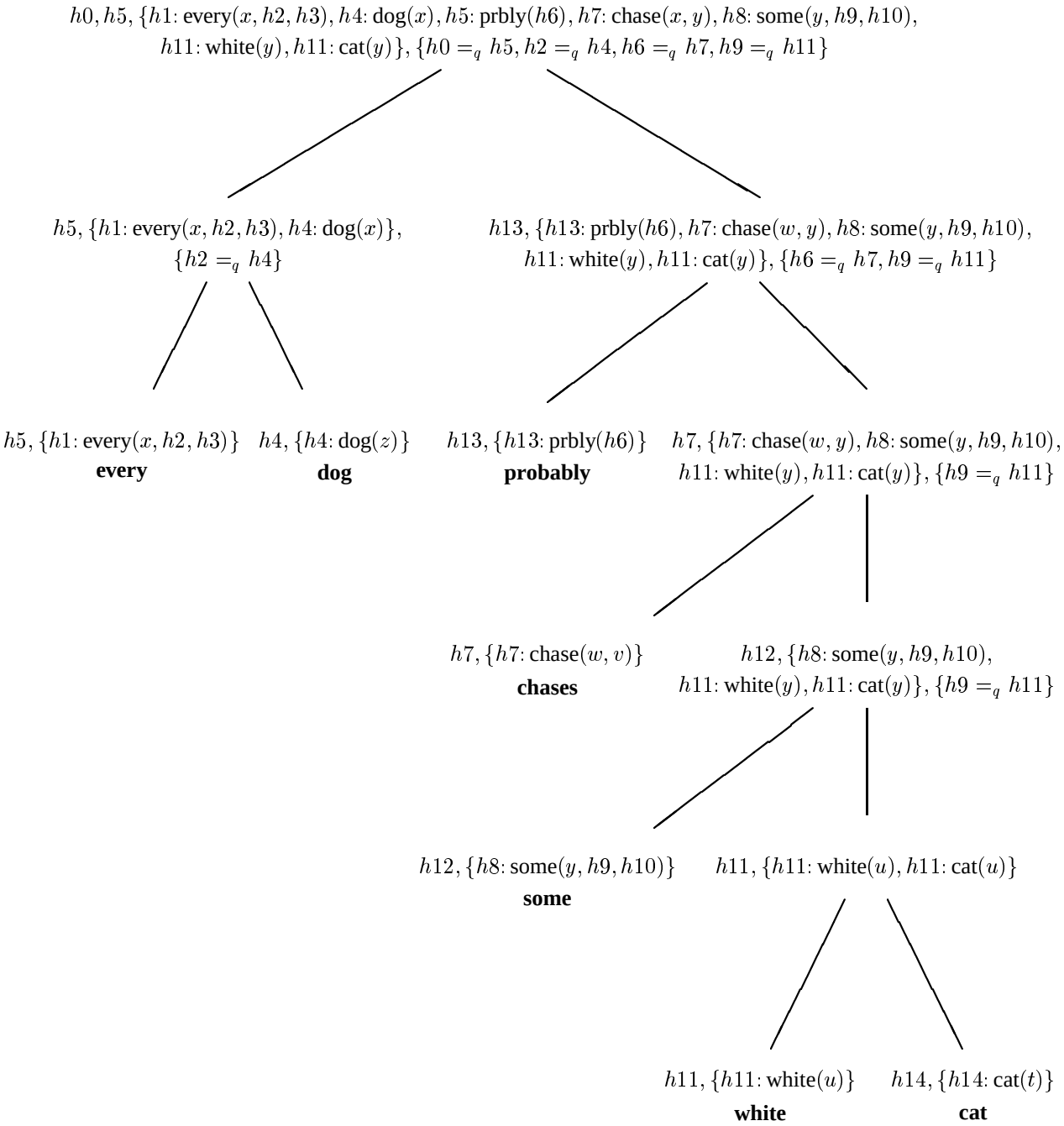


Figure 3: Example of composition. Empty handle constraints are omitted. The global top handle, h0, is omitted except on the root.

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 intuitively obvious that if a non-quantifier EP is not directly or indirectly stated to be qeq a scopal argument position or a quantifier's restriction, the only way it can get into that part of the structure when the scope is resolved is if it is carried there on the restriction of a quantifier. Quantifiers in their turn are restricted by the variable binding conditions and the requirement that they end up with something in their body. The combination of these conditions leads to various desirable properties: for instance, nothing can get inside the restriction of a quantifier unless it corresponds to something that is within the corresponding noun phrase syntactically. For instance, (27a) has the MRS shown in (27b), which has exactly the 5 scopes shown in (27c) to (27g).

(27) a every nephew of some fierce aunt saw a pony

b $\langle h0, h1, \{h2\colon \text{every}(x, h3, h4), h5\colon \text{nephew}(x, y), h6\colon \text{some}(y, h7, h8), h9\colon \text{aunt}(y), h9\colon \text{fierce}(y), h10\colon \text{see}(x, z), h11\colon \text{a}(z, h12, h$
$\{h1 =_q h10, h7 =_q h9, h3 =_q h5, h12 =_q h14\}\rangle$

c $\text{every}(x, \text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{nephew}(x, y)), \text{a}(z, \text{pony}(z), \text{see}(x, z)))$

d $\text{a}(z, \text{pony}(z), \text{every}(x, \text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{nephew}(x, y)), \text{see}(x, z)))$

e $\text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{every}(x, \text{nephew}(x, y), \text{a}(z, \text{pony}(z), \text{see}(x, z))))$

f $\text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{a}(z, \text{pony}(z), \text{every}(x, \text{nephew}(x, y), \text{see}(x, z))))$

g $\text{a}(z, \text{pony}(z), \text{some}(y, \text{fierce}(y) \wedge \text{aunt}(y), \text{every}(x, \text{nephew}(x, y), \text{see}(x, z))))$

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.

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 (28a), sketched in (28b), have to arise from a syntactic ambiguity.

(28) a Kim could not sleep

b could(not(sleep(kim)))
not(could(sleep(kim)))

This seems to be justified in this case, see e.g., Warner (1999). But there are other cases where the conditions should probably be relaxed, see §8.

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 grammar-independent: it is equally applicable to categorial grammars encoded in typed feature structure, for instance, although some of our encoding choices are motivated by following the existing HPSG tradition. In the next section we will move to discussing the use of MRS in a particular grammar, namely the English Resource Grammar developed by the LinGO project at CSLI (Flickinger *et al.*, in preparation, see also `http://hpsg.stanford.edu/hpsg/lingo.html`).

5.1 Basic structures

In feature structure frameworks, the semantic representation is a subpart of the structure which represents the word or phrase as a whole. Here we will just concentrate on that subpart and ignore the link with the syntax.

First we consider the encoding of EPs. We will assume that the EP's relation is encoded using the type of the feature structure, 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 instance, in (29) we show the structures for EPs for *dog* and *every*.

(29) a
$$\begin{bmatrix} \textbf{dog\_rel} \\ \text{HNDL } \textbf{handle} \\ \text{INST } \textbf{ref-ind} \end{bmatrix}$$

b
$$\begin{bmatrix} \textbf{every\_rel} \\ \text{HNDL } \textbf{handle} \\ \text{BV } \textbf{ref-ind} \\ \text{RESTR } \textbf{handle} \\ \text{BODY } \textbf{handle} \end{bmatrix}$$

As usual in typed feature structure formalism encodings of semantics, coindexation is used in MRS 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.

The use of types allows us to define a hierarchy of relations, with appropriate features according to their type. For instance, there will be a generic type **quant_rel**, which subsumes **every_rel**, **some_rel** and so on. To avoid confusing the names of generic types with specific relations, we will adopt the naming convention of preceding the latter with an underscore, for instance, **_every_rel**. This hierarchy of relations is useful in that it allows us to encode generalisations about semantic classes and linking, and also can support underspecification of relations. We will discuss this with reference to a specific grammar in §6.7. In this context we should mention that we avoid the use of very specific features for relations, such as CORPSE for the subject of **die**, This is traditional in the HPSG literature, but it is incompatible with stating generalisations involving the semantics. For instance, a separate linking rule would have to be stated for *die* and *snore*. 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 thematic roles such as ACT, UND (e.g., following Davis, 1996) or a semantically-bleached nomenclature, such as ARG1, ARG2. We will adopt the latter style here.

An MRS is defined as corresponding to a feature structure of type **mrs** with features TOP, LTOP, LZT and HCONS. TOP introduces the top handle, LTOP the local top. LZT is the feature that introduces the bag of EPs, which is implemented as a list in the feature structure representation. [11] 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 SC-ARG for the argument position handle and OUTSCPD for the label handle.

The MRS feature structure for the sentence *every dog probably sleeps* is shown along with its non-feature-structure equivalent in (30). For the sake of ease of comparison, the handle numbering has been made consistent between the two representations, though there is no number for the BODY of the **_every_rel** in the feature structure representation, because there is no coindexation, and conversely the variable which is represented as x in the non-FS form corresponds to the coindexation labeled with 8 in the FS form.

(30)
$$\begin{bmatrix} \textbf{mrs} \\ \text{TOP } \boxed{1}\ \textbf{handle} \\ \text{LTOP } \boxed{7}\ \textbf{handle} \\ \text{LZT} < \begin{bmatrix} \textbf{\_every\_rel} \\ \text{HNDL } \boxed{2}\ \textbf{handle} \\ \text{BV } \boxed{8}\ \textbf{ref-ind} \\ \text{RESTR } \boxed{4}\ \textbf{handle} \\ \text{BODY } \textbf{handle} \end{bmatrix}, \begin{bmatrix} \textbf{\_dog\_rel} \\ \text{HNDL } \boxed{6}\ \textbf{handle} \\ \text{INST } \boxed{3} \end{bmatrix}, \begin{bmatrix} \textbf{\_probably\_rel} \\ \text{HNDL } \boxed{7}\ \textbf{handle} \\ \text{ARG } \boxed{8} \end{bmatrix}, \begin{bmatrix} \textbf{\_sleep\_rel} \\ \text{HNDL } \boxed{9} \\ \text{ARG1 } \boxed{3} \end{bmatrix} > \\ \text{H-CONS} < \begin{bmatrix} \textbf{qeq} \\ \text{SC-ARG } \boxed{1} \\ \text{OUTSCPD } \boxed{7} \end{bmatrix}, \begin{bmatrix} \textbf{qeq} \\ \text{SC-ARG } \boxed{4} \\ \text{OUTSCPD } \boxed{6} \end{bmatrix}, \begin{bmatrix} \textbf{qeq} \\ \text{SC-ARG } \boxed{8} \\ \text{OUTSCPD } \boxed{9} \end{bmatrix} > \end{bmatrix}$$

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

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 (31).

(31) a
$$\begin{bmatrix} \text{RELN } \textbf{\_dog\_rel} \\ \text{HNDL } \textbf{handle} \\ \text{INST } \textbf{ref-ind} \end{bmatrix}$$

This allows the possibility of complex relations, such as the equivalent of Kind(beer), but it complicates the notion of a relation hierarchy. The second point is that instead of using coindexation between the values of HNDL and features corresponding to scopal arguments to get the effect of handles as we have discussed them so far, it would be possible to

[11] In earlier versions of MRS, the two features that we have called HNDL and LZT were called HANDEL and LISZT, with the justification that this is intended to be a form of compositional semantics. However, in view of the distress this has caused several readers, and the fact that we have not been able to find a composer with a name that remotely suggested HCONS, we have modified the feature names.

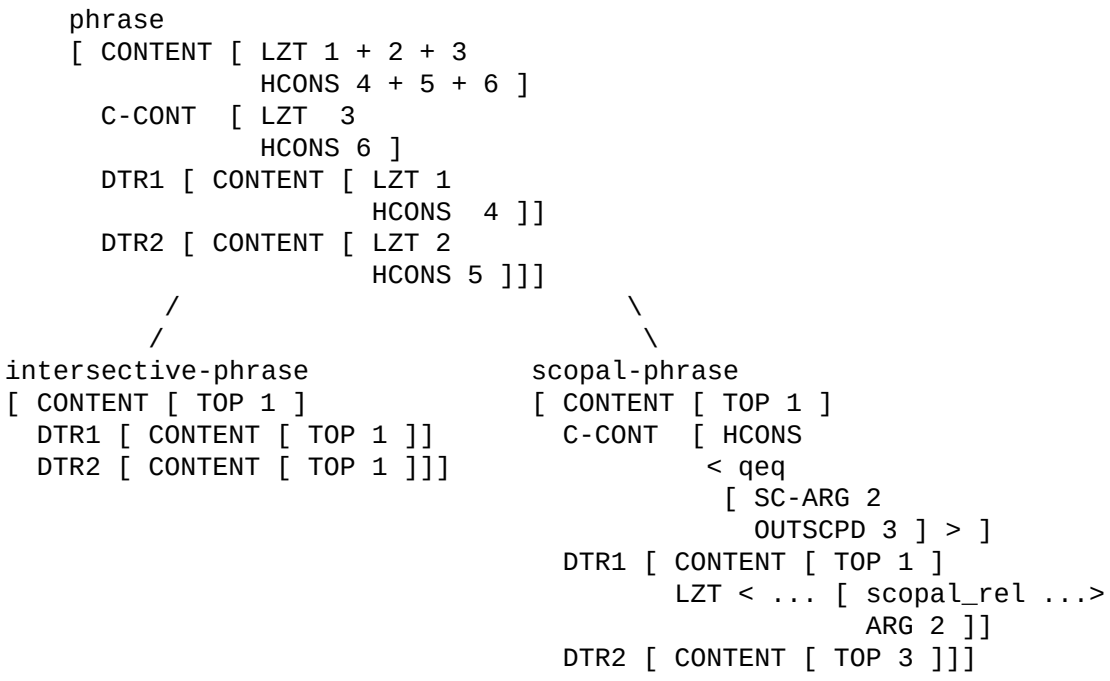


Figure 4: Illustrative constraint on composition

get a similar effect without the HNDL 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, which is unattractive both notationally and computationally.

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. We will not reiterate the lexical constraints, but the simple hierarchy shown in Figure 4 is one way of capturing the constraints on phrases, though this is intended to be illustrative rather than to correspond to the way the constraints would be implemented in any particular grammar. For simplicity, the rules are assumed to be binary, with daughters indicated by DTR1 and DTR2. DTR1 is the scoping daughter in **scopal-phrase**. The . . . notation in this structure is an abbreviation: the effect of distinguishing one particular EP argument would actually have to be captured by a feature. Unlike the previous description, Figure 4 allows for the phrase itself to contribute to the semantics: the feature C-CONT (constructional content) introduces an **mrs** which encodes the contribution of the phrase. C-CONT effectively is treated as if it were the semantics of a third daughter. For completeness, if we took the approach shown in Figure 4, we would have to provide another subtype of phrase for the case where the C-CONT key EP scoped over the daughters of the construction. C-CONT is discussed further in §6.5. The appends can be implemented by difference lists in those variants of feature structure formalisms which do not support append directly, although we ignore this complication and use the abbreviatory list notation here. We should empasize that Figure 4 is only one way of describing the constraints and is in many ways suboptimal: for instance it assumes that rules are divided into intersective and scopal, and in a lexicalist framework especially, it is desirable to allow for some rules which are general between these classes and achieve the same effect by means of lexical type distinctions.

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 shown the way that the coindexation propagates through the structure in this figure. We have shown the root condition as an extra level in the tree: this is not essential, but corresponds to the LinGO grammar we will discuss in §6.

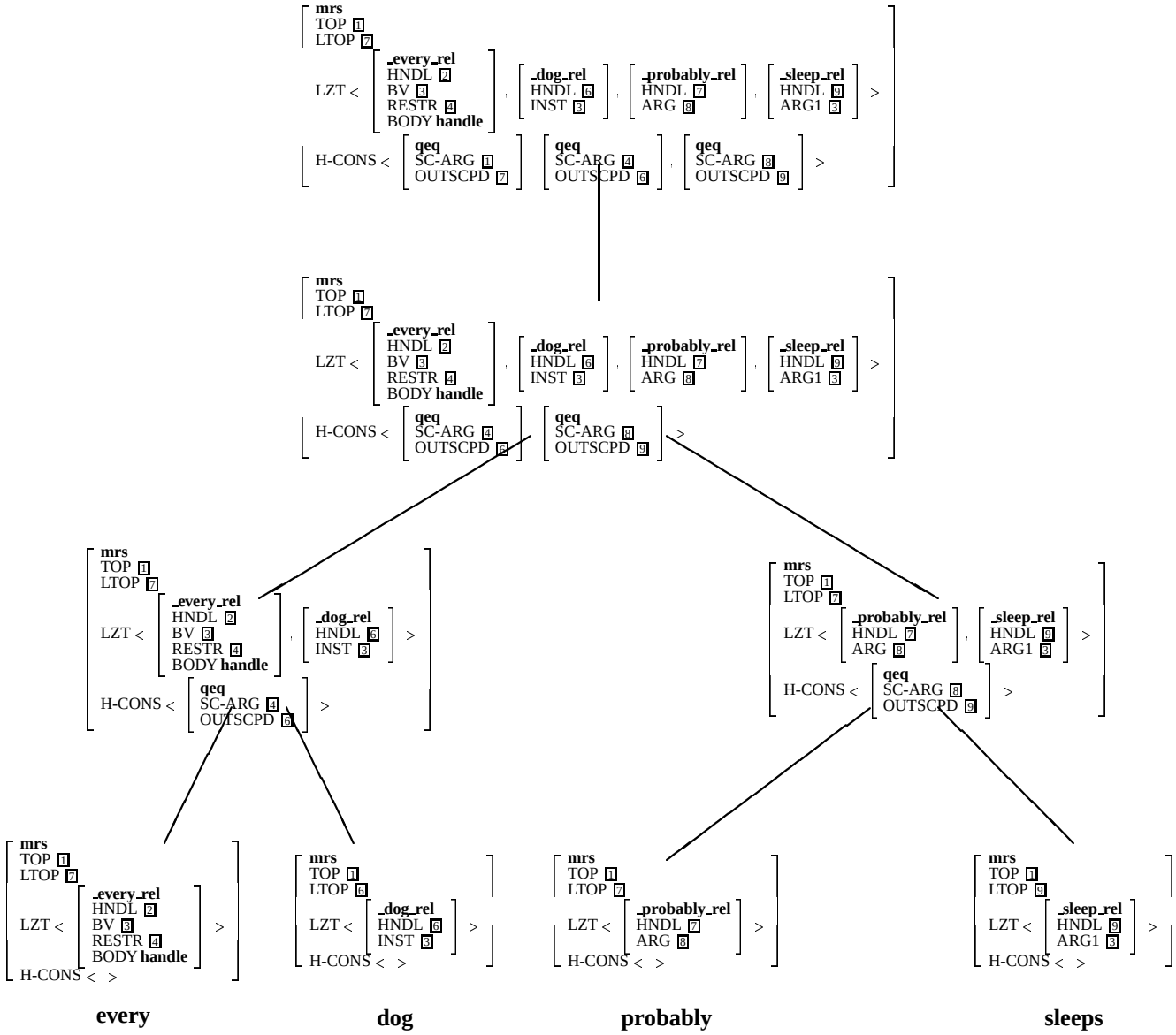


Figure 5: Example of composition using feature structures

5.3 Resolution

MRS was designed so that the process of scope-resolution could be formulated as a monotonic specialization of a feature structure. It is thus consistent with the general constraint-based approach assumed in HPSG. The discussion of linking in §4 basically holds in exactly 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.

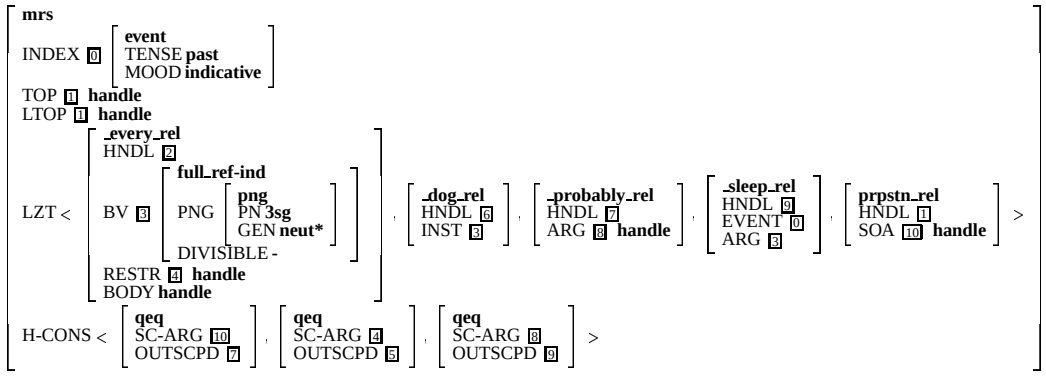
In principle, it would be possible to implement the instantiation of an underspecified MRS to produce scoped forms directly in a typed feature structure formalism which allowed recursive constraints. However to do this would be quite complex and there are very serious efficiency issues. The number of scopes of a sentence with n quantifiers and no other scopal elements is $n!$ (ignoring complications such as the case when one quantifier can be 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 determining scope. The situation is analogous to parsing and generation using typed feature structures: although it is possible to describe a grammar in a typed feature structure formalism in such a way that parsing and generation can be regarded as a form of resolution of recursive constraints, for efficiency most systems make use of special-purpose processing algorithms, such as variants of chart-parsing, rather than rely on general constraint resolution. And just as the use of a chart can avoid exponentiality in parsing, non-factorial algorithms can be designed for scoping. 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.*, in preparation, see also `http://hpsg.stanford.edu/hpsg/lingo.html`). We will do this by means of a series of examples, first repeating an example we've discussed already, and describing some additional grammar-specific aspects, and then going through a series of somewhat more complex cases.

6.1 A basic example

For our first simple example, consider the representation of the sentence *every dog probably sleeps* produced by the LinGO grammar which is shown in (32). This structure should be compared with (30).

(32)


The basic structures are as described in the previous section: the overall structure is of type **mrs** and it has slots for the LTOP, LZT and H-CONS as before. However, the global TOP is omitted as it is redundant for this grammar as we will explain shortly. We have also introduced a slot for INDEX: this is used in a similar way to a lambda variable, although because unification-based semantic representations are inherently very flexible in the way that semantic structures can be combined, the role of the INDEX is considerably more limited and is mainly relevant for modification. We will discuss this in more detail below (DRAFT — the section about indices is currently missing), but first go through some other differences.

Complex indices In the previous section, we assumed that the variables were represented by coindexed atomic values. In the LinGO grammar, as is usual in HPSG, these variables have their own substructure, which is used to

encode agreement for instance (Pollard and Sag, 1994:chapter 2). The the details of the encoding are not relevant to the MRS representation, however.

Event variables In this example, the value of the INDEX attribute is an event variable, which is coindexed with an argument of *die*. The LinGO grammar, in common with much other work on computational semantics, uses a neo-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 handles in MRS, and non-scopal adverbs, such as *quickly*, which are treated as event taking. This is analogous to the distinction between intersective and non-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.

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 past(e). A more semantically interesting representation might be *hold*(e, t), *precedes*$(t,$ *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 was 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 labored 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 that a more explicit representation can be derived.

Propositions The example shown in (32) contains an 'extra' EP (compared to (30)) which has a relation **prpstn_rel**. This conveys the information that the sentence is a proposition, rather than a question, for instance (questions are indicated by **_int_rel**). Such relations are generically referred to as *message* relations.

In general, message relations behave like normal scopal relations with respect to their arguments, 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 (we will see examples of messages occuring internally in an MRS structure below). 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 **prpstn_rel**. In fact this means that the feature TOP is redundant in this grammar, since the LTOP of a root structure is also the global top.

6.2 Intersective and scopal modifiers

Kasper (1996) investigates the internal semantic structure of modifiers, observing that previous analyses of modification in HPSG fail to account adequately for modifier phrases that contain modifiers of their own, as in his example (33).

(33) Congress approved the potentially controversial plan

He notes that on 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.

We can also describe the problem in terms of MRS semantics. In HPSG, the value of the syntactic feature MOD for an adjective phrase is instantiated by the noun modified. 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 for *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 lexically, via the MOD feature on the adjective, the result is wrong for *the potentially controversial plan*, since it means that *plan* and *controversial* share a label and thus when *potentially* is given scope over the EP for *controversial* it also ends 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 and is independently required in order to efficiently process modifiers for generation, as described by Carroll et al (1999). 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 the lexical coindexation of ltops for intersective adjectives and thus avoids the Kasper problem. In Figure 6 we show the composition of the MRS for *allegedly difficult problem*.

6.3 Sentential complements

DRAFT - section missing

6.4 Unbounded dependencies

DRAFT - section missing

6.5 Constructions

As we mentioned briefly above, MRS allows for the possibility that constructions may introduce EPs into the semantics. Constructions all have a 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, e.g., Carlson and Pelletier, 1995) and we do not propose to throw any light on the subject here, since we will simply assume that a generalized quantifier is involved, specifically **_udef_rel**, but say nothing about what it means.[12] 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 7 and an example of its application is in Figure 8. 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 and for compound nouns. 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*).

6.6 Conjunction

DRAFT — section missing

[12] To justify this a little, we would argue 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 go into a discussion here about whether it is appropriate to use a generalized quantifier, as opposed to some variety of kind reading for instance, since the point here is the mechanics of phrasal contributions to the semantics.

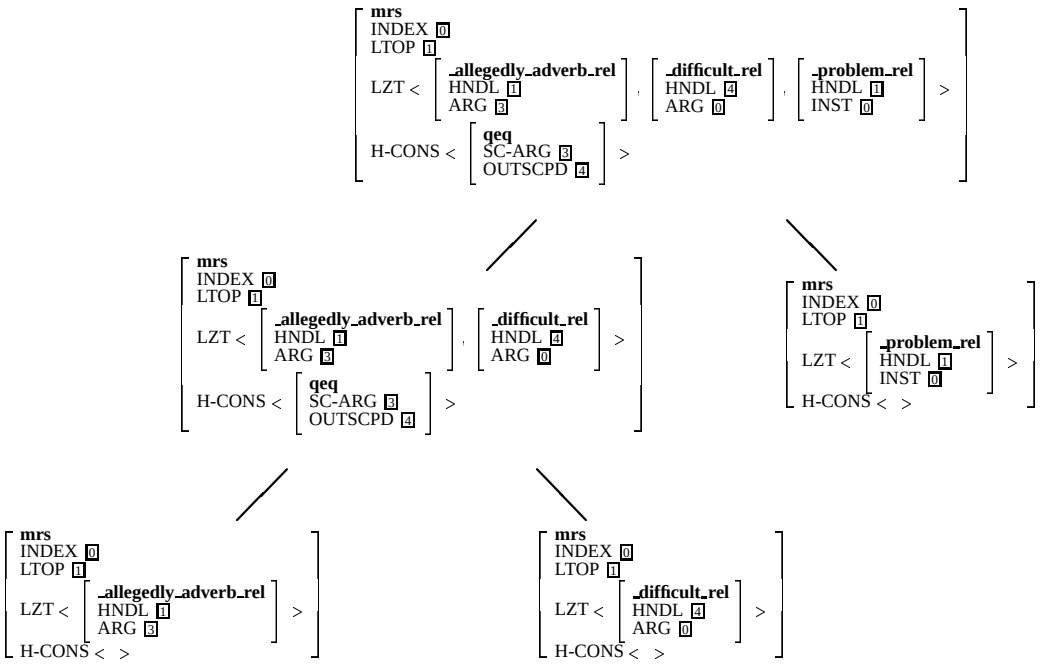


Figure 6: MRS for *allegedly difficult problem*. TOP is omitted throughout.

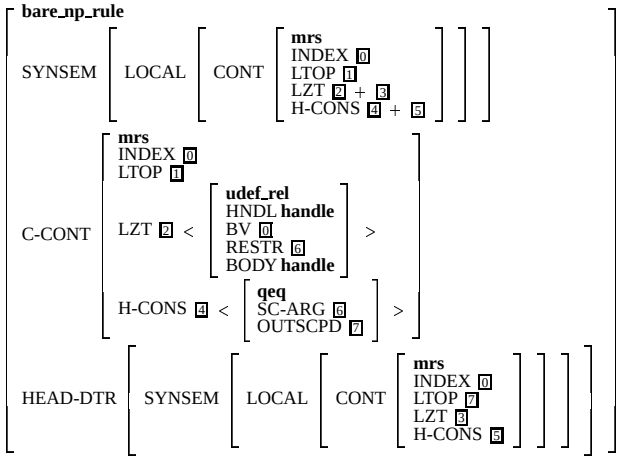


Figure 7: The bare-np rule

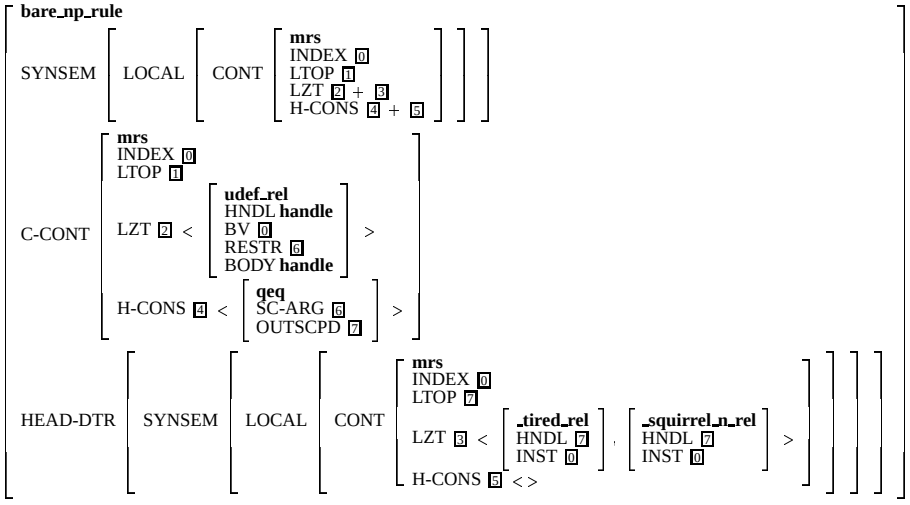


Figure 8: The bare-np rule applied to *tired squirrels*

6.7 Semantic selection and the relation hierarchy

DRAFT — section missing

7 Related Work

The idea of expressing meaning using a list of elements which correspond to individual morphemes has seemed attractive for 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 extremely ingenious, but the complexity of the relationship with more conventional approaches makes the representations somewhat difficult to follow and often incompatible with other approaches. As we discussed in §2, a form of flat semantics has been explicitly proposed for MT by Phillips (1993) and also 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. There has been considerable work on underspecification in semantics in the last few years (e.g., Alshawi and Crouch (1992), Reyle (1993) and the papers collected in van Deemter and Peters (1996) and in Crouch and Poesio (1996)).[13] The use of an underspecified intermediate representation in computational linguistics dates back at least to the LUNAR project (Woods et al., 1972) but 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). We believe that some of these approaches to the semantics of underspecified representations could also be compatible with MRS.

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 UDRT labels. However, MRS as described here makes use of a different style of scope constraint from UDRT. Our motivation was convenience of representation of the sort of constraints on scope which arise from syntax. 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 et al. (1996) describe LUD, which is similar to UDRS but more general, in that it can be used to describe underspecified semantic formalisms based on languages other than DRT. LUD distinguishes between *hole*s, which

[13] In this connection, one should also consult Esther König's 'Underspecification Database', whose URL is:
`http://www.ims.uni-stuttgart.de/projekte/sfb/b3/b3-db.html`

are analogous to our argument handles, and *label*s, which correspond to our use of labels to distinguish particular elementary predications. Unlike UDRS and MRS, LUD assumes unique labels on the EPs. From our perspective however, perhaps the main difference between LUD and MRS is that LUD is implemented as a distinct semantic construction component in a grammar with a GB influenced syntactic component. MRS, in contrast, was developed for HPSGs with a tighter syntax-semantics interface, and is explicitly designed for use in 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 at IBM Heidelberg. UMRS is described in Egg and Lebeth (1995, 1996) and Egg (1998). Abb et al (1996) discuss the use of UMRS in semantic transfer (also see Copestake et al (1995) and Copestake (1995) for semantic transfer using MRS). MRS, UMRS and LUD have all been used on the Verbmobil machine translation project. They are sufficiently similar that interconversion of logical forms between the representations is practically reasonably straightforward. We believe that these three representations all go a considerable way towards meeting the criteria for computational semantics that we listed in the introduction.

Riehemann (1996) describes a treatment of idioms which crucially utilizes MRS. The 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 which is insensitive to irrelevant syntactic distinctions. Egg (1998) describes an approach to wh-questions in UMRS. This work indicates that flat semantics may have advantages in theoretical linguistic research with no computational component.

Finally we should mention that Alshawi (e.g., 1996) and others have argued for a much more radical approach to semantics for natural language processing systems which avoids explicit semantic representation altogether and treats (annotated) natural language strings directly as expressions of an underspecified representation. Alshawi suggests that context can be provided by the state of the language processor, rather than in an explicit representation. One motivation for this work is making automatic acquisition easier. While we would argue that it is preferable to maintain a separate declarative semantic representation such as MRS (at least for manually developed grammars), we think that Alshawi's work highlights the need to consider the computational utilization of semantic formalisms and the desirability of keeping representations close to natural language.

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.

Acknowledgments

This material is based upon work supported by the National Science Foundation under grant number IRI-9612682 and by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant FKZ:01iV401.

We are grateful for the comments and suggestions of colleagues working on Verbmobil and at CSLI. We would like to acknowledge in particular the valuable contribution that Harry Bunt made to the development of the ideas presented in this paper. However, as usual, the responsibility for the contents of this study lies with the authors.

References

Abb, B., B. Buschbeck-Wolf and C. Tschernitschek (1996) 'Abstraction and underspecification in semantic transfer', *Proceedings of the Second Conference of the Association for Machine Translation in the Americas (AMTA-96)*, Montreal, pp. 56–65.

Alshawi, H. (1996) 'Head automata and bilingual tiling: translation with minimal representations', *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-94)*, Santa Cruz, CA, pp. 167–176.

Alshawi, H. and R. Crouch (1992) 'Monotonic Semantic Interpretation', *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-92)*, Newark, NJ, pp. 32-39.

Bos, J., B. Gambaeck, C. Lieske, Y. Mori, M. Pinkal, K. Worm (1996) 'Compositional Semantics in Verbmobil', *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen.

Carlson, G.N. and F.J. Pelletier (eds) (1995) *The Generic Book*, University of Chicago Press, Chicago.

Carroll, J., A. Copestake, D. Flickinger and V. Poznanski (1999) 'An Efficient Chart Generator for (Semi-)Lexicalist Grammars', *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG'99)*, Toulouse, pp. 86–95.

Cooper, R. (1975) 'Montague's Semantic Theory and Transformational Syntax', Doctoral dissertation, University of Massachusetts at Amherst.

Cooper, R. (1983) *Quantification and Syntactic Theory*, Reidel, Dordrecht.

Copestake, A., D. Flickinger, R. Malouf, S. Riehemann, I.A. Sag (1995) 'Translation using Minimal Recursion Semantics', *Proceedings of the The Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*, Leuven, Belgium.

Copestake, A. (1995) 'Semantic transfer for Verbmobil', ACQUILEX II working paper 62 and Verbmobil report 93.

Copestake, A. (1996) 'Underspecification and defaults in lexical semantic representation', Paper presented at 'From underspecification to interpretation', Max-Planck Research Group workshop, Berlin.

Crouch, R. and M. Poesio (1996) 'Discourse interpretation with underspecified representations', Reader for ESSLII-96, Prague.

Davis, A. (1996) 'Linking and the hierarchical lexicon', Doctoral dissertation, Stanford University.

van Deemter, K. and S. Peters (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**, 37–82.

Egg, M. and K. Lebeth (1995) 'Semantic underspecification and modifier attachment', Intergrative Ansätze in der Computerlinguistik. Beiträge zur 5. Fachtagung für Computerlinguistik der DGfS.

Egg, M. and K. Lebeth (1996) 'Semantic interpretation in HPSG', Presented at the Third International Conference on HPSG, Marseilles, France.

Frank, A. and U. Reyle (1994) 'Principle based semantics for HPSG', Arbeitspapiere des Sonderforschungsbereichs 340, University of Stuttgart.

Hobbs, J. (1983) 'An improper treatment of quantification in ordinary English', *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', *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL-85)*, Chigaco, IL, pp. 61-69.

Kamp, H. and U. Reyle (1993) *From Discourse to Logic: an introduction to modeltheoretic semantics, formal logic and Discourse Representation Theory*, Kluwer Academic Publishers, Dordrecht, Germany.

Kasper, Robert (1996) 'Semantics of Recursive Modification', ms. Ohio State University.

Kay, M. (1970) 'From semantics to syntax' in M. Bierwisch and K.E. Heidorn (ed.), *Progress in Linguistics*, Mouton, The Hague, pp. 114–126.

Kay, M. (1996) 'Chart Generation', *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, Santa Cruz, CA, pp. 200–204.

Landsbergen, J. (1987) 'Isomorphic Grammars and their use in the ROSETTA Translation System' in M. King (ed.), *Machine Translation Today: The State of the Art*, Edinburgh University Press, pp. 351–372.

Partee, B., A. ter Meulen and R.E. Wall (1993) *Mathematical methods in linguistics*, Kluwer academic publishers, Dordrecht.

Phillips, J.D. (1993) 'Generation of text from logical formulae', *Machine Translation*, **8 (4)**, 209–235.

Pollard, C. and I.A. Sag (1994) *Head-driven phrase structure grammar*, Chicago University Press, Chicago.

Pollard, C. and E.J. Yoo (1996) 'A Unified Theory of Scope for Quantifiers and *Wh*-Phrases', ms. Ohio State University. To appear in *Journal of Linguistics*.

Poznanski, V., J.L. Beaven and P. Whitelock (1995) 'An efficient generation algorithm for lexicalist MT', *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, Cambridge, Mass., pp. 261–267.

Reyle, U. (1993) 'Dealing with ambiguities by underspecification', *Journal of Semantics*, **10**, 123–179.

Riehemann, S. (1996) 'Idiomatic constructions in HPSG', ms. Stanford University, available via `ftp://ftp-csli.stanford.edu/linguistics/idioms-sr.ps`.

Shieber, S.M. (1993) 'The problem of logical form equivalence', *Computational Linguistics*, **19 (1)**, 179–190.

Trujillo, I.A. (1995) 'Lexicalist Machine Translation of Spatial Prepositions', PhD dissertation, University of Cambridge.

Warner, A. (1999) 'English auxiliaries without lexical rules' in R. Borsley (ed.), *The nature and function of syntactic categories*, Academic Press, New York.

Whitelock, P. (1992) 'Shake-and-Bake translation', *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France.

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