Dependency and (R)MRS

Ann Copestake

aac@cl.cam.ac.uk
December 9, 2008

1 Introduction

Note: for current purposes, this document lacks a proper introduction, in that it assumes readers know about MRS and RMRS.

The MRS and RMRS languages are successful in allowing grammar writers to construct underspecified compositional semantics for a variety of languages. They can be used with feature structure formalisms, but also in a separate semantic module with other styles of syntax. RMRS can be used with syntactic approaches that have much less lexical information than grammars which have previously been used to construct compositional representations. However, there is a major disadvantage from the perspective of using these languages for human annotation: they are verbose and difficult to read. This same disadvantage also makes them unsuitable for direct use with machine learning approaches: compositional semantic features can be extracted from (R)MRS, but the structures cannot be used directly. The verbosity also makes some types of manipulation of (R)MRS unnecessarily complex algorithmically. These properties also apply (to at least some extent) to other existing compositional semantic representations.

The aim of this work is to look at whether (R)MRS (as used in the DELPH-IN grammars) can be interconverted with a representation that has more of the properties of a dependency represenatation. The main reason for looking at dependency structures is to find an encoding that minimises redundancy. A dependency structure utilises relatively simple connections between elements in an utterance, while a compositional semantic representation appears much more complex. This interconversion is also theoretically interesting, in that it provides a link between compositional semantics, constituency and dependency. It should be stressed that the intention is not that the dependency structure be thought of as an alternative form of semantics: the dependency links do not have a direct semantic interpretation.[1]

It is clear that an (R)MRS can be converted to some type of dependencies: Stephan Oepen developed code to do this some years ago which is incorporated in the LKB system. However, the question here is whether we can come up with an encoding which is formally equivalent so that that the dependencies can be converted back into the original (R)MRS in all cases. It is clear that it would be possible to do this for individual grammars, but this would mean the representation was subject to change as the grammar changed and that annotating with dependencies would require detailed knowledge of the grammar. What we require is that the interconversion depend on general properties which hold of a group of grammars, which can be stated formally and which will be stable as the grammars develop. We also want to be able to define a 1:1 relationship, such that we can define a deterministic procedure for converting between a (R)MRS and a corresponding DMRS.

The conclusion of the work discussed here is that specific properties of the semantics in DELPH-IN grammars do allow MRS representations to be encoded in a surprisingly succinct dependency style representation, which I will refer to as DMRS. We can have some confidence that these grammar properties are theoretically reasonable, because they emerged from a style of compositional semantics which has been developed after experience with multiple grammars in a range of languages and linguistic frameworks. These were developed without any idea of a translation into dependency structure. Not all styles of semantic representation would allow this straightforward transformation: the crucial attributes of DELPH-IN (R)MRS which make it possible are a) the decision to use a relatively 'surfacy' style of representation b) quantifer scope underspecification c) the use of a simple algebra in composition. A fourth essential property, which has emerged from the grammars but has not been discussed previously, is the notion of 'characteristic' variables, made more precise below, in §2.4. This is important in that it makes it possible to define a single DMRS as the canonical one corresponding to any RMRS.

This paper will proceed by taking an initial simple example of an RMRS and showing a step-by-step deterministic conversion to a dependency structure in DMRS, justifying the claim that the conversion preserves information at each

[1]The links actually correspond to operations in the algebra for MRS composition, as will be discussed in more detail at the end of this paper.

stage. I will then give an outline of the RMRS to DMRS conversion algorithm and of DMRS to RMRS conversion. The discussion will then move on to consider some issues which were not relevant to the initial example.

2 Compositional semantics and dependency structure: a worked example

Consider the sentence:

Most deaf cats are white.

Assuming that *most* is a generalised quantifier, the representation for this could be written as:[2]

_most_q(x, _deaf_a_1(e1,x) & _cat_n_1(x), _white_a_1(e2,x))

This representation uses the established DELPH-IN convention where predicate names which correspond to lexemes consist of an underscore, followed by the lexeme stem, followed by another underscore and a letter giving a coarse-grained sense distinction based on the part-of-speech (e.g., '_n' for nouns). A further sense distinction may be indicated by another underscore followed by a number of letters indicating the sense (e.g., '_cat_n_1'). One point about this analysis which should be noted is the lack of an explicit representation of the copula which in turn necessitates the event variables on the adjectival predicates. In a full representation, the event variables have associated sorts, indicating tense (among other things). Thus the fact that this sentence is present tense is indicated by the sort associated with e2. Sorts will be omitted for now, but sorts in DMRS are discussed in §4.6.

The following structure shows the MRS for the sentence which a compositional analysis would be expected to generate: [3]

$LTOP : l3$
$l0$: _most_q$(x, h1, h2)$,
$l1$: _deaf_a_1$(e1, x)$,
$l1$: _cat_n_1(x),
$l3$: _white_a_1$(e2, x)$,
$h1 qeq l1$

The structure corresponds to the generalised quantifier representation shown previously. It is underspecified, in that the BODY of the quantifer is not specified at all and the RSTR is only specified via a qeq relationship. This example has only one resolved form.

One point which is important in what follows: resolving MRSs to give fully scoped forms involves replacing qeq relationships with hole-label equalities. In the current form of MRS, resolution from the structure produced by a complete syntactc analysis may not involve any other operations: in particular, two distinct labels may not become equivalent via resolution. Similarly, when MRSs are output from a complete syntactic analysis, all variable equalities are taken to be known. Anaphoric relationships are not treated by equating variables but by an anaphoric variable link. Formally, we assume that there are a set of inequality relationships between all distinct labels and variables (or in the version of MRS with explicit equalities, between all labels and variables that are not specified to be equal). This assumption is made explicit in generation from an MRS.

This structure is further disassembled to give the RMRS. Note that this version of RMRS uses anchors to associate the arguments with the appropriate predicate. Note also that each predicate has exactly one argument which is directly associated with it: the reason for defining RMRS in this way was that all predicates have at least one such argument and the basic sort of such arguments is always known on the basis of the predicate class. e.g., an adjectival predicate

[2] In fact, *most* is not a quantifier in the current version of the ERG, and *white* is not a (simple) adjective, but I'm expecting these decisions will be changed!

[3] The formal syntax of RMRS and MRS are defined in XML.

always has a first argument of type 'e'.

$$
\begin{array}{l}
LTOP: l3 \\
l0\text{: } a0\text{: \_most\_q}(x), \\
a0\text{: RSTR}(h1), \\
a0\text{: BODY}(h2), \\
l1\text{: } a1\text{: \_deaf\_a\_1}(e1), \\
a1\text{: ARG1}(x), \\
l1\text{: } a2\text{: \_cat\_n\_1}(x), \\
l3\text{: } a3\text{: \_white\_a\_1}(e2), \\
a3\text{: ARG1}(x), h1qeql1
\end{array}
$$

An equivalent representation is to make all equalities specific, which turns out to be useful in defining the interconversion between DMRS and RMRS:

$$
\begin{array}{l}
LTOP: l3 \\
l0\text{: } a0\text{: \_most\_q}(x0), \\
a0\text{: RSTR}(h1), \\
a0\text{: BODY}(h2), \\
l1\text{: } a1\text{: \_deaf\_a\_1}(e1), \\
a1\text{: ARG1}(x1), \\
l2\text{: } a2\text{: \_cat\_n\_1}(x2), \\
l3\text{: } a3\text{: \_white\_a\_1}(e2), \\
a3\text{: ARG1}(x3), \\
h1qeql1 \\
x0 = x1, x1 = x2, x1 = x3, l1 = l2
\end{array}
$$

The structure below shows that the representation used for processing associates character positions with the elementary predications. These are usually omitted for clarity.

$$
\begin{array}{l}
LTOP: l3 \\
l0\text{: } a0\text{: \_most\_q} < 0, 3 > (x), \\
a0\text{: RSTR}(h1), \\
a0\text{: BODY}(h2), \\
l1\text{: } a1\text{: \_deaf\_a\_1} < 5, 11 > (e1), \\
a1\text{: ARG1}(x), \\
l1\text{: } a2\text{: \_cat\_n\_1} < 13, 23 > (x), \\
l3\text{: } a3\text{: \_white\_a\_1} < 29, 35 > (e2), \\
a3\text{: ARG1}(x), \\
h1qeql1
\end{array}
$$

Constrast this with a 'typical' dependency representation which might have links as shown in Figure 1. Note that such a representation does not distinguish between the given sentence 'Most deaf cats are white' and 'Most white cats are deaf'.[4] The crucial piece of missing information arises from the constituent structure that is not preserved in Figure 1 that *deaf* is inside the Nbar and thus should be inside the restriction of the quantifier while *white* is not.

Although the (R)MRS representations look more complex, the dependency representation shown in Figure 2 can be automatically derived from the (R)MRS and converted back into an equivalent structure (ignoring sorts on variables, which are discussed in §4.6).[5] In the next sections, I will give a step-by-step explanation of this example.

2.1 Step 1: conversion of RMRS into a graph

As a first attempt at a dependency structure, we construct a graph which has nodes consisting of the RMRS predicates, the variables and the labels. The labels are related to the corresponding elementary predication by arcs annotated LBL.

[4] As far as I can tell from brief web browsing, the first statement is true for congenital deafness in cats, while the second is false, although most white cats with blue eyes are deaf.

[5] One proviso, for examples where there are uninstantiated optional arguments, the availability of a SEM-I (see below) is required for complete reconstruction of the (R)MRS structure, although a valid RMRS structure which preserves all the information about links between items can be constructed without a SEM-I.

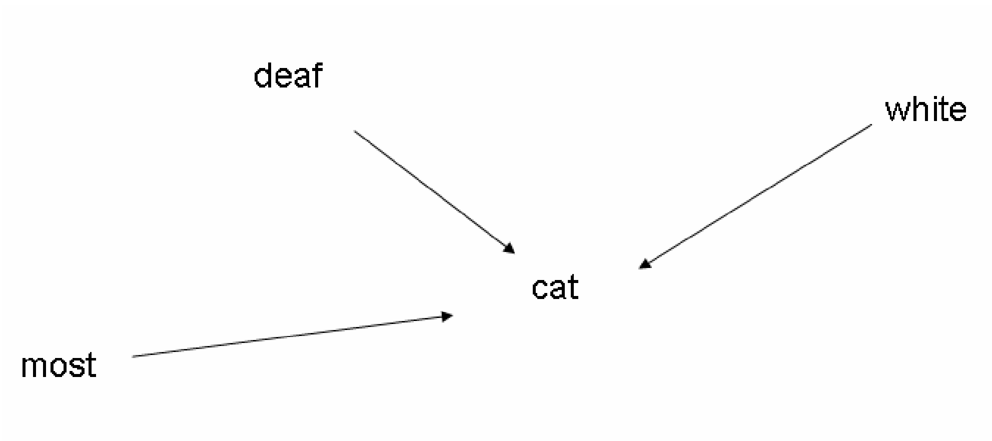


Figure 1: Schematic dependency structure

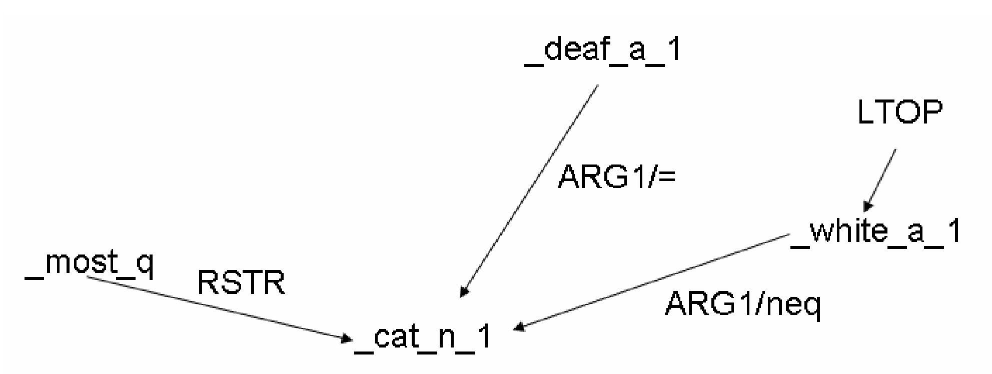


Figure 2: DMRS structure

The variables in the RMRS which were included directly in the elementary predication are linked to it via ARG0 links, with the exception of the quantifiers, where the link is notated BV (for convenience of discussion). The remaining arcs in the graph relate predicates to variables: they are annotated with the corresponding RMRS argument. The anchors are not needed, since the arguments are connected directly to the elementary predicates (i.e., the anchors are only necessary in RMRS because the elements are written independently). It should be clear it is possible to convert this graph back to an equivalent RMRS, though possibly with different names for the anchors. Note that the highly

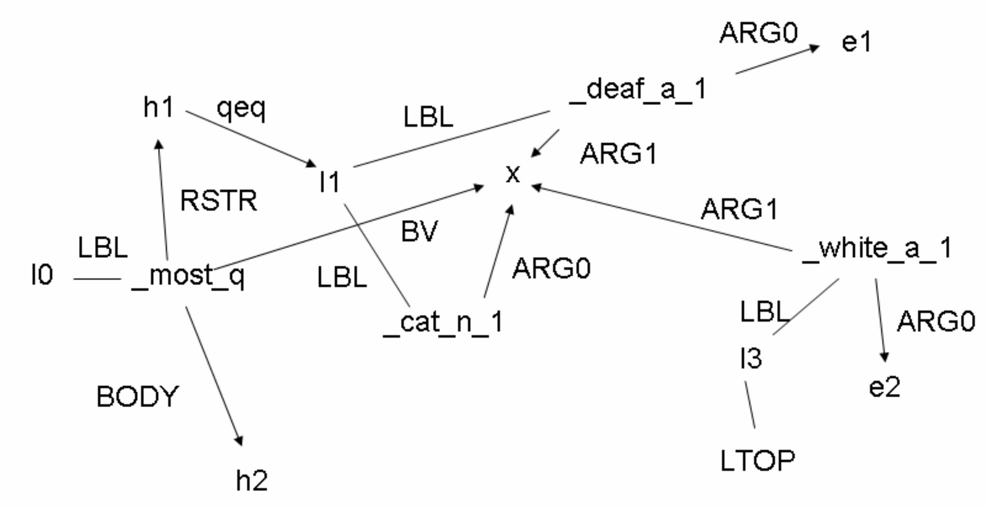


Figure 3: RMRS graph

lexical nature of the (R)MRS predicates is crucial in ultimately producing something that resembles a conventional dependency representation: in DMRS, we will end up with nodes that correspond to predicates, most of which will be lexical (as all are here). Subsequent steps are concerned with demonstrating that we can remove the variables in this graph and drastically simplify the link structure.

2.2 Step 2: removal of unshared variables

To reduce the complexity of the graph, we retain only those variables (and labels) which are shared between two or more elementary predications. This is shown in Figure 4. This step is lossy, in that to reconstruct the original RMRS

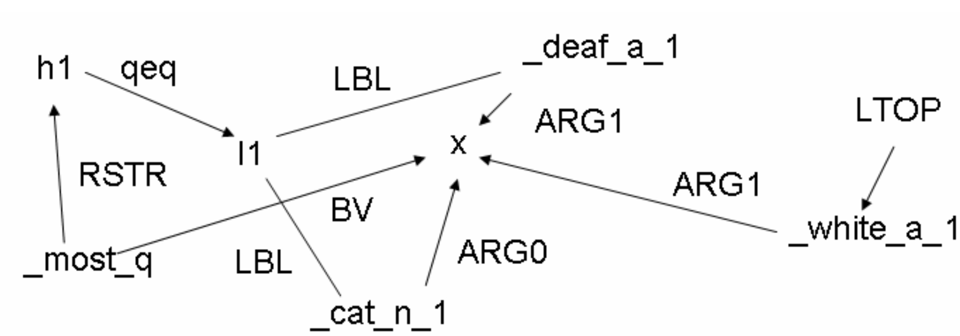


Figure 4: RMRS graph, omitting variables which are not shared

we need information about the total set of arguments appropriate to the elementary predication, but this information is lexical and does not depend on the individual utterance being analysed. The MRS language is fixed-arity. RMRSs may omit arguments and be well-formed, but conversion from RMRS to MRS when arguments are omitted requires a semantic interface (SEM-I), which among other things, is used in DELPH-IN grammars to record all the arguments appropriate for a particular predicate. Given the SEM-I, we can reconstruct the ARG0 arguments for the adjectives, and the BODY argument for the quantifier. For this particular example, the non-shared variables could all be inferred based on the predicate class, but the SEM-I would be needed for cases such as *eat*, which has an optional ARG2.

Note that LTOP is now linked directly to a predicate rather than a label, because l3 has been removed.

2.3 Step 3: replacing qeqs

The MRS algebra specifies that all scopal links are qeqs. Thus, in any case where we have a hole-taking argument position (such as RSTR here), where in the full representation the argument has a hole value whih is qeq a label, we can remove the hole and the qeq link and simply make the argument directly relate to the label. This is shown in Figure 5. There is a complication here, in that in grammars such as the ERG, there are a few exceptions where a scopal

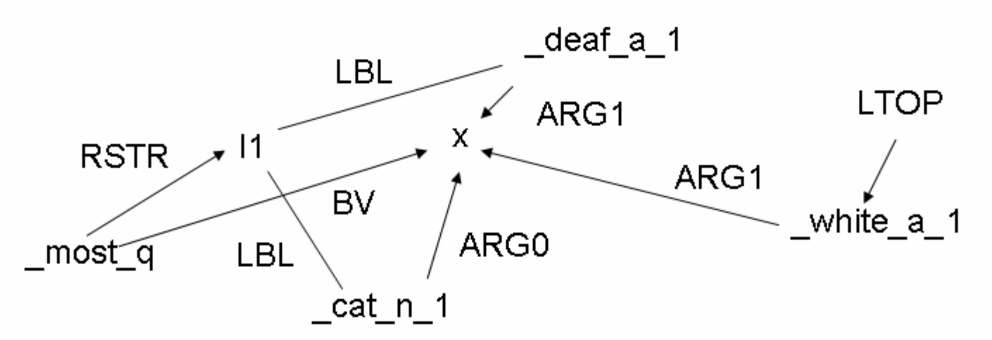


Figure 5: RMRS graph, removing qeqs

link is made directly, rather than via a qeq. If we take this step without somehow noting the cases to which it does not apply, we would incorrectly generate qeqs when going back from a DMRS structure to an RMRS. However, these exceptions apply to a few specific constructional predicates, which can be treated specially. This will be discussed in §4.3.

2.4 Step 4: characteristic variables

At this point, the graph still contains the variable x and the label l1. In general, such graphs will still contain shared variables and labels, though holes will have been removed. To produce a dependency style representation, we need to remove the remaining variables and labels, so we just have links between the predicates. The crucial step in doing this deterministically is establishing a unique predicate that can be said to 'own' any variable. One way of doing this would be to create a dominance hierarchy based on syntax (e.g., syntactically an adjective modifies a noun and thus the noun can be said to dominate), but this would be external to the semantic structures we are working with, and specifying a complete hierarchy leads to complex rules. Instead we will make use of a principle that has already emerged informally in the DELPH-IN grammars, which is that any variable will be 'characteristic' for exactly one predication. If the procedure described so far is followed, these variables are always linked to the corresponding predicate via ARG0 links. Thus we can remove the variables and ARG0 links from the graph in favour of links from the non-characteristic predicates directly to the predicate which 'owned' the variable via its ARG0 link. Note that the BV variable was directly associated with the quantifier in the RMRS but is characteristic to the noun, not the quantifier: this is discussed further in §4.1. A graph illustrating this state is shown in Figure 6. For this figure, there are two points associated with each elementary predicate in the graph: the position after the predicate corresponds to the characteristic variables, and the position in front corresponds to labels. There is a non-directional arc indicating label equality between _deaf_a_1 and _cat_n_1.

The characteristic variable idea gives results which are intuitive, since nouns own a object-denoting variable (in fact this is the only argument for most nouns) and verbs own an event variable. If the characteristic variable principle holds true, it is always possible to remove variables from the RMRS graph to produce a DMRS deterministically. If there are variables for which the owning predicate cannot be established, this would not be possible, and we would have to fall back on an externally specified notion of ownership.

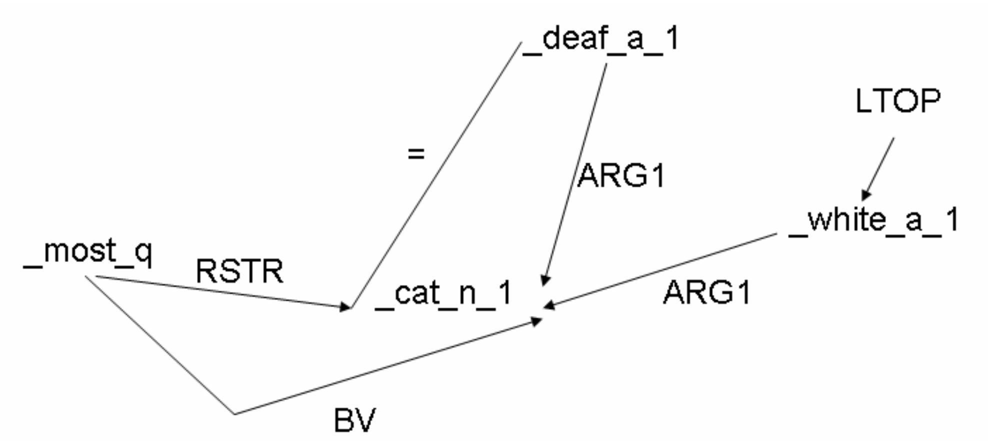


Figure 6: Graph with no variables

Conversely, considering the transformation of a DMRS to an RMRS, we note that this is always possible (modulo variable naming) because a variable will be introduced for each case where there is a dependency. All predicates have associated labels, of course, and equality between them is indicated by the arcs.

There is one temporary cheat in this figure — I have shown the RSTR label from the quantifier pointing to the noun predicate rather than the adjective, but have not explained the rationale for the choice. Intuitively it follows the fact that the BV is the characteristic variable of the noun rather than the adjective, as discussed further below. In general, the unique target of a hole argument can be established indirectly via the characteristic variable hierarchy. The equality links between labels are non-directional, but the link merging step discussed in the next section will merge them with the directional variable links in most cases.

2.5 Step 5: consolidating arcs, removing label/variable distinction on target predications.

The final major step is to observe that it is notationally unnecessary to have dual arcs between predicates indicating their label and variable relationships separately if we adopt a more complex arc labelling scheme. The structure shown in Figure 7 illustrates this. Note that the expedient of having separate label/variable positions for the previous figure has now been removed, so there is a 'clean' dependency structure with links between the predicates. The labels on the arcs have two parts: the first is the argument label, as before, while the second (after the slash) indicates the relationship between the predications in the (R)MRS scope tree. The possibilities for label relationships between predications are:

1. h (qeq relationship)
2. = (label equality)
3. neq (label non-equality)
4. nothing (underspecified label relationships, discussed in §5 below)

Logically the BV arc should also be /h, but that arc will be removed in the next step, so we ignore that complication.

This labelling scheme arises directly from the MRS algebra. In the algebra, combination is either scopal (in which case there is a qeq relationship between some hole and the label of the argument) or intersective, in which case labels are shared. The third, 'no relationship', case might appear to violate the MRS algebra, but arises here because of the special treatment of quantifiers and the decision to treat nouns as owning the characteristic variable. Further discussion of this interaction is in §4.1. Note that the neq case makes explicit the non-equality of labels which is implicit in the RMRS.

It should be clear that it is possible to reconstruct the dual links from this structure. Creating the dependency structure with the dual links will always be possible if the inventory of links is sufficient. The case of the non-qeq relationships in the ERG, alluded to earlier, means one additional link type will be necessary. A further class of link is required where a label equality holds without there being an argument relationship of any type between the predicates.

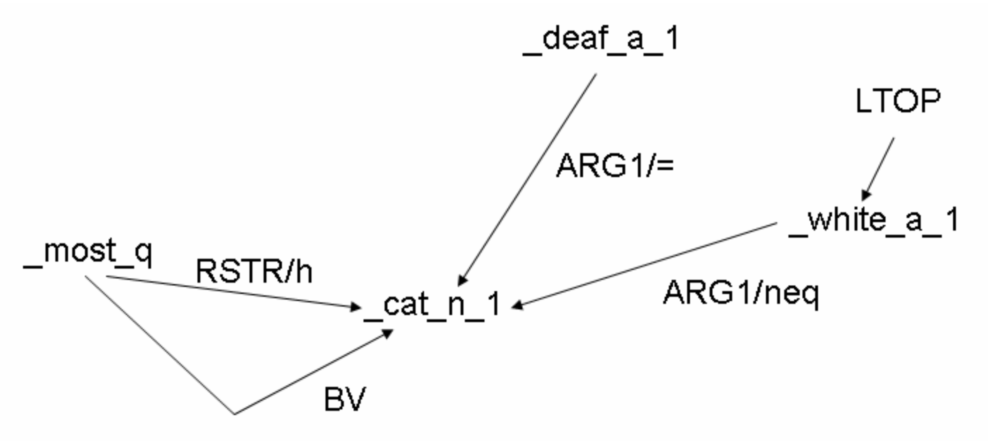


Figure 7: Consolidated arc labels

This is notated /= and has no directionality. This arises in the ERG in some relatively unusual examples, some of which are discussed below, in §4.4.

2.6 Step 6: removal of further redundancy

There are some further steps that can be taken to remove redundancy. The BV link and the RSTR argument link can be combined: the resulting link is annotated RSTR. The /h is unnecessary, since RSTR arguments are always of this type. This gives us the structure shown in Fig 8, which is repeated from Fig 2. The possibility of combining the BV

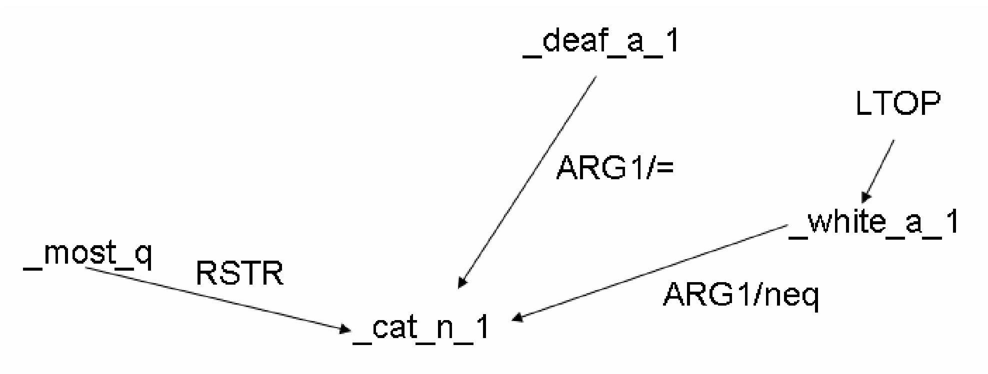


Figure 8: Final DMRS structure (repeated from Fig 2)

and the RSTR link arises because the BV always corresponds to the characteristic variable associated with the nominal predicate which is the semantic head of the Nbar. This predicate will always have a label which is the value of RSTR, although this label may be shared by other predicates. The possibility of combination of the RSTR and the BV link justifies the fact that the combined link points to the nominal rather than the adjectival predicate in the example: i.e., we assume that in the case where there is a choice between two or more locations for the target of an 'h' type link, a decision can be made on the basis of the location of an associated characteristic variable.

The removal of the /h is possible because the RSTR label is unambiguously a qeq relationship, unlike ARG1, ARG2 etc labels, which vary.

3 DMRS

3.1 DMRS structure summary

A DMRS is a connected acyclic graph. All nodes are RMRS predicate symbols with the exception that there is exactly one node labelled LTOP with one or more arcs which link it to other node(s). Arcs may have an associated direction (the connectedness requirement applies when the arcs are taken as undirected — the direction is perhaps best though of as an annotation rather than part of the graph structure). The LTOP arc(s) are unlabelled, all other directed arcs have labels of the form ARG, ARG/h, ARG/= or ARG/neq, where ARG is taken from the RMRS inventory of arguments. There may also be undirected /= arcs.

3.2 RMRS to DMRS conversion algorithm

To transform an RMRS into a DMRS:

1. Each predicate in the RMRS becomes a node in the DMRS graph. If there is a constant (CARG) value of the predicate, this is added to the node: e.g., named_rel/"Sandy". (This assumes there is only one constant per predicate.)

2. Each variable y in the RMRS is associated with a unique predicate which owns it (call this predicate charp(y)).

3. For every non-characteristic argument A (except BV arguments) in the RMRS which is anchored to a predicate P and has a variable value y, such that charp(y) is P', an arc exists in the DMRS from P to P'. The arcs wil be referred to as **variable arcs**. The preslash label on that arc is equal to the argument type of A (e.g., ARG1). If the labels of P and P' are equated, then the post-slash label is =. If P has a hole argument which is qeq the label of P', then the post-slash label is h. If P and P' have labels which are unrelated in the RMRS, the post-slash label is neq.

4. For every case where there are predicates P and P' in the RMRS which have equal labels and the equalities are not fully represented by equalities attached to variable arcs (taking transitive connections into account), there is a non-directional /= link between P and P'.

   (Note: the need for /= links is apparently very rare, but the above specification would lead to an excessive number of links if there are more than two predicates with an unrepresented shared label. Whether this actually ever happens is unclear, but, if it does, the following definition may be preferable (this is what is implemented in the code, currently). It amounts to the same thing for the two predicate case.

   For every case where a set S of predicates in the RMRS have equal labels and the linkages are not fully represented by equalities on variable arcs (taking transitive connections into account), we define an ordering P_0 to P_n on S by character position of the predicates. There is a non-directional /= link between P_0 and each other predicate in P_1 to P_n considered successively which is not linked by previous operations.)

5. If a label lb labels a set of predicates, P_1 to P_n, then the **head predicates** in that set are defined as those which have no outgoing variable arcs to other elements in the set.

6. If the RMRS LTOP has value lb labelling a set of predicates, then the LTOP node in the DMRS is linked to each head predicate in that set.

7. For every argument A which is anchored to a predicate P and has a hole value h, such that h is qeq a label lb which labels a set of predicates S:

   (a) If P has exactly one argument A' with a value y such that $P' =$ charp(y) is a member of S, then there is an ARG/h arc from P to P', where ARG is the argument type of A. (Note that this covers the RSTR case because there will be a BV argument.)

   (b) Otherwise, there are ARG/h arcs from P to each of the head predicates in S.

   This allows for multiplication of arcs if there's no way of making it deterministic according to semantic information. I haven't used the character position strategy here, because the head of the arc is always fixed, so the arcs numbers will be small, if indeed there are any cases which don't et resolved by the above.

8. If the preslash label on the arc uniquely determines the postslash label, the slash and the postslash label may be omitted. (e.g., we write RSTR, rather than RSTR/h).

3.3 DMRS to RMRS conversion

To transform a DMRS into an RMRS (with explicit equalities), the following steps are required:

1. Label each DMRS predicate with a distinct label and anchor.
2. Associate each predicate with all possible arguments given by SEM-I lookup (including the characteristic variables and the BVs of quantifiers), keeping all argument names unique.
3. Split CARG values from the predicate.
4. For each ARG/= link between two predicates P and P':
   (a) add an equality between the labels of the predicates P and P'
   (b) add an equality between the ARG variable of P and the characteristic variable of P' (the ARG variable is the value of the argument labelled ARG). (If there is no ARG variable associated with P, or if it is a different type from the characteristic variable of P', the DMRS is ill-formed according to the SEM-I.)
5. For each /= link between two predicates P and P', add an equality between the labels of the predicates P and P'.
6. For ARG/h link between predicates P and P' (including arcs with no post-slash which are known to be of type h, such as RSTR), add a qeq between the ARG hole of P and the label of P'. (If there is no ARG hole associated with P, the DMRS is ill-formed according to the SEM-I.)
7. For each ARG/neq link between predicates P and P', add an equality between the ARG variable of P and the characteristic variable of P'. (If there is no ARG variable associated with P, or if it is a different type from the characteristic variable of P', the DMRS is ill-formed according to the SEM-I.)
8. For each RSTR link between P and P', add an equality between the BV argument of P and the characteristic variable of P'.
9. Make the LTOP of the RMRS the label of the predicate linked to LTOP in the DMRS. If multiple predicates are linked to LTOP, equate their labels.

The following shows these steps applied to our example DMRS:

1. Label each predicate with a distinct label and anchor.

$l0$: $a0$: _most_q
$l1$: $a1$: _deaf_a_1
$l2$: $a2$: _cat_n_1
$l3$: $a3$: _white_a_1,

2. Add SEM-I arguments:

$l0$: $a0$: _most_q$(x0)$,
$a0$: RSTR$(h1)$,
$a0$: BODY$(h2)$,
$l1$: $a1$: _deaf_a_1$(e1)$,
$a1$: ARG1$(x1)$,
$l2$: $a2$: _cat_n_1$(x2)$,
$l3$: $a3$: _white_a_1$(e2)$,
$a3$: ARG1$(x3)$,

3. Add the h type links (here, just RSTR):

$l0$: $a0$: _most_q$(x0)$,
$a0$: RSTR$(h1)$,
$a0$: BODY$(h2)$,
$l1$: $a1$: _deaf_a_1$(e1)$,
$a1$: ARG1$(x1)$,
$l2$: $a2$: _cat_n_1$(x2)$,
$l3$: $a3$: _white_a_1$(e2)$,
$a3$: ARG1$(x3)$,
$h1qeql2$

4. Add the ARG/= links:

$l0$: $a0$: _most_q$(x0)$,
$a0$: RSTR$(h1)$,
$a0$: BODY$(h2)$,
$l1$: $a1$: _deaf_a_1$(e1)$,
$a1$: ARG1$(x1)$,
$l2$: $a2$: _cat_n_1$(x2)$,
$l3$: $a3$: _white_a_1$(e2)$,
$a3$: ARG1$(x3)$,
$h1qeql2$
$l1 = l2$,
$x1 = x2$

5. Add the ARG/neq links:

$l0$: $a0$: _most_q$(x0)$,
$a0$: RSTR$(h1)$,
$a0$: BODY$(h2)$,
$l1$: $a1$: _deaf_a_1$(e1)$,
$a1$: ARG1$(x1)$,
$l2$: $a2$: _cat_n_1$(x2)$,
$l3$: $a3$: _white_a_1$(e2)$,
$a3$: ARG1$(x3)$,
$h1qeql2$
$l1 = l2$,
$x1 = x2, x3 = x2$

6. Add BV links for the RSTRs:

$l0$: $a0$: _most_q$(x0)$,
$a0$: RSTR$(h1)$,
$a0$: BODY$(h2)$,
$l1$: $a1$: _deaf_a_1$(e1)$,
$a1$: ARG1$(x1)$,
$l2$: $a2$: _cat_n_1$(x2)$,
$l3$: $a3$: _white_a_1$(e2)$,
$a3$: ARG1$(x3)$,
$h1qeql2$
$l1 = l2$,
$x1 = x2, x3 = x2, x0 = x2$

7. Add the LTOP:

$$
\begin{array}{l}
LTOP : l3 \\
l0{:}\, a0{:}\ \_\text{most\_q}(x0), \\
a0{:}\, \text{RSTR}(h1), \\
a0{:}\, \text{BODY}(h2), \\
l1{:}\, a1{:}\ \_\text{deaf\_a\_1}(e1), \\
a1{:}\, \text{ARG1}(x1), \\
l2{:}\, a2{:}\ \_\text{cat\_n\_1}(x2), \\
l3{:}\, a3{:}\ \_\text{white\_a\_1}(e2), \\
a3{:}\, \text{ARG1}(x3), \\
h1 qeq l2 \\
l1 = l2, \\
x1 = x2, x3 = x2, x0 = x2
\end{array}
$$

It's worth noting that the DMRS to RMRS conversion is relatively simple compared to the converse, essentially because there's no concern for minimisation or canonicalisation of equality links.

In the absence of a SEM-I, a valid RMRS can still be produced, but the procedure is a little more complex to state and the result could lack some arguments. The x/e sortal distinction will also be missing.

4 Details and discussion

4.1 Dependency structure and the MRS treatment of scope

At this point, it is useful to consider what is behind the need to have a three way distinction between label behaviours. The links in the dependency structure essentially correspond to operations in the semantic algebra. In general, the dependency is written so that the functor (i.e., the semantic head) points to its argument. According to the (R)MRS algebra, the two possibilities for combination of a semantic head and its argument are a) for the semantic head to have a hole which is qeq the label of its argument (scopal arguments) or b) for the semantic head to share a label with its argument (intersective combination). These possibilities correspond to ARG/h and ARG/= respectively. The reason why we also have ARG/neq is the way that quantified NPs behave in MRS. In order to allow quantifiers to float, the algebra specifies that the label of a quantifier is not equated with the label of the structure that contains the quantifier relation. The semantic head of the NP is the quantifier (not the N), so the label of a quantified NP is not identified with the label of any relation. Composition with an NP argument technically falls into the intersective/label-equal situation in the algebra, but because the NP label is isolated from the components of the NP, this has no effect on the structure.

We have chosen to associate the variable which corresponds to the entity with the noun. In fact, regardless of arguments about characteristic variables, the only viable option appears to be for the target of the arc to be taken as the nominal predicate, regardless of whether the composition in the original structure involved the noun, the NBar or the NP. Since we are dealing with dependencies, there is no representation of the phrase which allows us to distinguish between these cases. Making the determiner predicate the target when the composition involves a (quantified) NP and the nominal predicate the target otherwise will thus not work, because it would violate the requirement that each variable has a unique 'home'. Logically, there is also the possibility of making the determiner the owner of the variable and thus the target uniformly, but this does not work for NPs which have no quantifier (e.g., pronouns). Furthermore it is counterintuitive to link constituents which are syntactically nominal modifiers (adjectives etc) to the determiner rather than the noun and any such representation require additional links to identify the labels of intersective Nbar modifiers (attributive adjectives etc) with the noun. Note that this label equivalence must be preserved in order to obtain the correct restriction of the quantifier. In other words, we need a representation which allows for both Nbar and NP arguments, such that we can reconstruct the label equalities between NBar elements without incorrectly equating the label of a noun in a quantified NP. Adjectives, PPs and verbs may all appear in contexts where they combine with an NBar (attributive position, relative clauses) and also in contexts where their argument is an NP (predicative position, main clause verbs). Hence the distinction between NP and Nbar cannot be made on the basis of the part-of-speech of the arc origin. The solution adopted here means that the Nbar case is annotated with ARG/= arcs, while the NP case has ARG/neq arcs. Thus the ARG/neq arcs are playing the same role as the unattached labels in the orginal (R)MRS.

It is also possible for /neq links to occur when a non-local semantic argument relationship is involved, as discussed in §6.1.

One further point is the nature of the BV argument to the generalised quantifiers. It is treated differently here in that a) the variable associated with it is not treated as the characteristic variable of the quantifier, despite the fact it is incorporated in the RMRS elementary predication and b) there is no arc for BV in the dependency representation. The intuition behind this is that the BV argument is not a genuine semantic argument at all, but a notational convenience. Semantically, quantifiers take two arguments which in typed lambda calculus would each be of type $\langle e, t \rangle$: i.e., quantifiers are of type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$. Generalised quantifiers are sometimes written as relationships between two lambda expressions: e.g., $\text{every}'(\lambda x[\text{dog}'(x)], \lambda y[\text{bark}'(y)])$ However, for convenience in grammar writing, we want to be able to equate indices of subject and verb phrase directly. Hence the frequently used alternative which was adopted in MRS, which is equivalent to: $\text{every}'(x, \text{dog}'(x), \text{bark}'(x))$ Thus the BV 'argument' is really just a convenient way of reifying the lambda variable and the fact that it has no direct equivalent in the dependency representation is desirable.

4.2 LTOP

The LTOP label of the dependency representation is the LTOP as defined by the MRS algebra. The rules for constructing the DMRS are specified so that only a subset of the relations with the LTOP handel will become linked to LTOP in the DMRS.

4.3 Constructions

For some simple sentences, such as that shown in Fig 1, there is a one-to-one correspondance between words and elementary predications, and hence a one-to-one correspondance between words and nodes in the dependency representation. However, this is not the usual case.

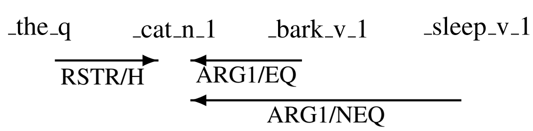
all those who wrote oppose the changes.

Constructions without qeqs.

special cases: in a few instances, the ERG does not obey the algebra conditions and hardwires a relationship between a hole and a label rather than having a qeq. This includes conjunction of VPs (though not sentences). The hope is that these cases can be recognised and enumerated, and given special arc labels. For instance, CONJ-L-VP would be interpreted as a hardwired link, whereas CONJ-L-S would be a qeq. These sort of special cases may turn out to be a problem, although the difference between a qeq link and a direct link is unlikely to have much impact.

CARGs

4.4 Relative clauses



The dog whose ball I chased barked.

The dog the owner of whom I chased barked.

The dog who I chased the owner of barked.

4.5 The role of the SEM-I

The representation described only takes care of the interrelationships between elementary predications.

The SEM-I will sometimes determine whether an arc should be ARGn/= vs ARGn/h.

4.6 Sorts on variables: tense, number etc

5 Underspecification

Underspecification of compound bracketing, PP-attachment etc appears possible.

6 Syntactic dependencies versus semantic dependencies

6.1 Non-local semantic operations

Nbar label identification retains a necessary element of constituent structure.