Unterspezifikation in der Semantik
Minimal Recursion Semantics (MRS)

Laura Kallmeyer
Heinrich-Heine-Universität Düsseldorf

Wintersemester 2011/2012

Overview

1. Introduction
2. Why flat semantics?
3. MRS: Idea
4. MRS: Definition
5. MRS scope constraints

[Copestake et al., 2005]

Introduction

Miminal Recursion Semantics [Copestake et al., 2005] is a semantics framework that

- is a meta-language for the description of semantic structures in an underlying object language,
- is computationally tractable,
- assumes a flat semantic structure,
- allows underspecification for scope phenomena, and
- can be realised in the form of typed feature structures which allows then to use it in HPSG.

Why flat semantics? (1)

Main motivation: intersective modifiers

(1) fierce black cat

meaning of the adjectives and the noun:

1. $\lambda P.\lambda x[\mathit{fierce}'(x) \wedge P(x)]$
2. $\lambda P.\lambda x[\mathit{black}'(x) \wedge P(x)]$
3. $\lambda x[\mathit{cat}'(x)]$

The composition, following a standard analysis, yields

$$\begin{aligned}[[\text{fierce black cat}]] &= \lambda x[\mathit{black}'(x) \wedge \lambda x[\mathit{fierce}'(x) \wedge \lambda x[\mathit{cat}'(x)](x)](x)] \\ &= \lambda x[\mathit{black}'(x) \wedge [\mathit{fierce}'(x) \wedge [\mathit{cat}'(x)]]]\end{aligned}$$

Why flat semantics? (2)

$$\lambda x[black'(x) \wedge [fierce'(x) \wedge [cat'(x)]]]$$

is equivalent to

$$\lambda x[cat'(x) \wedge [fierce'(x) \wedge [black'(x)]]]$$

and to a series of other groupings.

⇒ one would like to have a single representations for all these bracketing combinations.

Ex.: Spanish

(2) gato negro y feroz

yiels $cat'(x) \wedge [black'(x) \wedge fierce'(x)]$ which would not allow the generation of (1) in a system where we assume that translation pairs have the same semantic structures.

Why flat semantics? (3)

Possible solution:

$$\lambda x. \bigwedge \{black'(x), fierce'(x), cat'(x)\}$$

This is however not enough structure. We need a little bit of structure in order to deal with scope:

(3) every black cat is fierce

Here, *black′* and *cat′* are in the restriction of the quantifier, which is not the case for *fierce′*. Therefore the three predications should not be part of the same unordered set.

MRS: Idea (1)

(4) every big white horse sleeps

Predicate logic in prefix notation:

$$every'(x, \bigwedge(big'(x), \bigwedge(white'(x), horse'(x))), sleep'(x))$$

Step 1: Removing embeddings inside conjunctions:

$$every'(x, \bigwedge(big'(x), white'(x), horse'(x)), sleep'(x))$$

Step 2: Simplify the notation: Group the elements of conjunctions into bags (i.e., sets where multiple occurrences of elements are possible) of elementary predications that are interpreted as conjunctions:

$$every'(x, \{big'(x), white'(x), horse'(x)\}, sleep'(x))$$

MRS: Idea (2)

We want to be able to talk about subexpressions. Therefore we introduce handles (correspond to holes and labels/to node variables in dominance constraints).

$$h0 : every'(x, h1, h2)$$
$$h1 : big'(x), h1 : white'(x), h1 : horse'(x)$$
$$h2 : sleep'(x)$$

- Extension: Underspecification.
- Idea: We can use handles not only for labelling purposes (see above) but also as holes, i.e., as arguments inside elementary predications.
- In addition, scope constraints between handles are introduced.

MRS: Idea (3)

(5) every dog chases some white cat

$$
\begin{array}{l}
h1 : every'(x, h2, h3) \\
h2 : dog'(x) \\
h4 : some'(y, h5, h6) \\
h5 : white'(y), h5 : cat'(y) \\
h7 : chase'(x, y)
\end{array}
$$

Conditions on desambiguating (i.e., on equations of handles, “plugging”):

- All arguments must be filled (i.e., no handle in argument position may be left underspecified).
- An elementary predication (EP) must not belong to more than one argument position.

MRS: Idea (4)

$$
\begin{array}{l}
h1 : every'(x, h2, h3) \\
h2 : dog'(x) \\
h4 : some'(y, h5, h6) \\
h5 : white'(y), h5 : cat'(y) \\
h7 : chase'(x, y)
\end{array}
$$

$h3 =?$, $h6 =?$.

Possible values for $h3$: $h4, h7$. Possible values for $h6$: $h1, h7$.

If $h3 = h7$, then necessarily $h6 = h1$.

If $h6 = h7$, then necessarily $h3 = h4$.

$h3 = h4$ and $h6 = h1$ would yield a non-tree structure which should be excluded by the system as well.

MRS: Definition (1)

An elementary predication EP consists of

- a handle h,
- a relation R,
- a list of argument variables in the object language $x_1, \dots, x_k$ that are arguments of R,
- a list of handles $h1, \dots, hm$ that represent scope arguments of R.

Notation: $h : R(x_1, \dots, x_k, h1, \dots, hm)$

An EP conjunction is a bag of EPs that have all the same handle.

MRS: Definition (2)

Scope relations implied by MRS stuctures:

- An EP E outscopes another EP E', if the label handle of E' is an argument of E.
- We extend this outscopes relation by its reflexive transitive closure, as usual, so that we obtain a partial order.
- Furthermore, we extend the outscoping relation to entire EP conjunctions: An EP conjunction K outscopes another EP conjunction K', if the label handle of K' is an argument of one of the EPs in K.
- Finally, we extend the outscoping relation to the handles of EPs in the obvious way.

MRS: Definition (3)

In order to define our MRS structures, we add an additional top handle and scope constraints to the EPs.

A MRS structure is a tuple $\langle GT, R, C\rangle$ where GT is the top handle, R is a bag of EPs and C is a bag of handle constraints such that there is no handle that outscopes GT.

$$\left\langle h0, \left\{ \begin{array}{l} h1: \mathit{every}'(x, h2, h3), \\ h2: \mathit{dog}'(x), \\ h4: \mathit{some}'(y, h5, h6), \\ h5: \mathit{white}'(y), h5: \mathit{cat}'(y), \\ h7: \mathit{chase}'(x, y) \end{array} \right\}, \emptyset \right\rangle$$

We will treat the specific form of scope constraints used in MRS later.

MRS: Definition (4)

A scope-resolved MRS structure is an MRS structure that satisfies the following conditions:

1. The MRS structure forms a tree when considering handles as nodes and outscoping as dominance.
2. The top handle and all handle arguments are identified with (i.e., equal to) an EP label.
3. All constraints are satisfied.

$$\left\langle h1, \left\{ \begin{array}{l} h1: \mathit{every}'(x, h2, h3), \\ h2: \mathit{dog}'(x), \\ h3: \mathit{some}'(y, h5, h6), \\ h5: \mathit{white}'(y), h5: \mathit{cat}'(y), \\ h6: \mathit{chase}'(x, y) \end{array} \right\}, \emptyset \right\rangle$$

MRS: Definition (6)

- A MRS M' link-subsumes an MRS M if is can be obtained from M by adding additional equations between handles.
- A well-formed MRS structures is an MRS structure that link-subsumes at least one scope-resolved MRS strucure.

Interesting forms of linking (= equating handles):

- equating argument handles ("holes") with label handles;
- equating label handles to form a larger EP conjunction,
- equating the top handle with a label handle.

MRS scope constraints (1)

The definition of resolved MRS structures is not enough to make sure we obtain exactly the scope readings we want.

(6) every nephew of some famous politician runs

$$\left\langle h0, \left\{ \begin{array}{l} h1: \mathit{every}'(x, h2, h3), \\ h4: \mathit{nephew\_of}'(x, y), \\ h5: \mathit{some}'(y, h6, h7), \\ h6: \mathit{politician}'(y), h6: \mathit{famous}'(y), \\ h8: \mathit{run}'(x) \end{array} \right\}, \emptyset \right\rangle$$

This does not capture that $h4: \mathit{nephew\_of}'(x, y)$ is in the restriction of every', i.e., $h2$ must outscope $h4$.

MRS scope constraints (2)

MRS uses *qeq* constraints (written $=_q$) that stand for equality modulo quantifiers. They always relate holes (argument handles) to label handles.

Intuition: $h =_q l$ means

- either h gets filled by l ($h = l$)
- or one or more quantifiers 'float in' between h and l. I.e., h is filled by the label of a quantifier such that the body of this quantifier is either filled by l or, again, by a second quantifier and so on.

Crucial difference to dominance constraints: Only quantifiers can get into a *qeq* relation, embedding the lower label within their body!

MRS scope constraints (3)

With the *qeq* constraints, we specifiy those handles that have to be part of the restriction of a quantifier and we say explicitely that the top hole must outscope the smallest proposition containing all non-quantifier material.

(7) every nephew of some famous politician runs

$$\left\langle h0, \left\{ \begin{array}{l} h1 : every'(x, h2, h3), \\ h4 : nephew\_of'(x, y), \\ h5 : some'(y, h6, h7), \\ h6 : politician'(y), h6 : famous'(y), \\ h8 : run'(x) \end{array} \right\}, \left\{ \begin{array}{l} h0 =_q h8 \\ h2 =_q h4, \\ h7 =_q h4, \end{array} \right\} \right\rangle$$

MRS scope constraints (4)

Exmple involving an intensional adverb tha takes scope as well but that is no quantifying NP:

(8) every dog probably chases some white cat

$$\left\langle h0, \left\{ \begin{array}{l} h1 : every'(x, h2, h3), \\ h8 : dog'(x), \\ h4 : some'(y, h5, h6), \\ h9 : white'(y), h9 : cat'(y), \\ h7 : chase'(x, y), \\ h10 : probably'(h11) \end{array} \right\}, \left\{ \begin{array}{l} h0 =_q h10, \\ h2 =_q h8, \\ h5 =_q h9, \\ h11 =_q h7 \end{array} \right\} \right\rangle$$

MRS scope constraints (5)

$$\left\langle h0, \left\{ \begin{array}{l} h1 : every'(x, h2, h3), \\ h8 : dog'(x), \\ h4 : some'(y, h5, h6), \\ h9 : white'(y), h9 : cat'(y), \\ h7 : chase'(x, y), \\ h10 : probably'(h11) \end{array} \right\}, \left\{ \begin{array}{l} h0 =_q h10, \\ h2 =_q h8, \\ h5 =_q h9, \\ h11 =_q h7 \end{array} \right\} \right\rangle$$

This signifies that the top handle outscopes *probably* which in turn outscopes the *chase* proposition. In between, the two quantifiers can come in any way they like.

But: if $h3 =_q h7$ is added, the order *every* > *probably* is excluded. So, even though *probably* is also a quantifier (over situations or possible worlds) it is not treated as such.

References

[Copestake et al., 2005] Copestake, A., Flickinger, D., Pollard, C., and Sag, I. A. (2005). Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3:281–332.