Morphosyntax-Semantics Interface in Lexicalist Theories: Minimal Recursion Semantics (MRS) in a Nutshell

Julia Neu and Valia Kordoni

Email: kordoni@coli.uni-sb.de

WWW: http://www.coli.uni-sb.de/~kordoni/ss03/

Introduction-Motivation

- in natural language processing there has always been a need for compact semantic representations which can express, among others, scope ambiguities, avoiding at the same time disjunctive specifications of many different possible readings;
- Minimal Recursion Semantics (MRS; Copestake, Flickinger, Sag, and Pollard (1999), Copestake (2002)) provides a framework which supports such semantic representations;
- Minimal Recoursion Semantics is not considered to be a semantic theory itself, but a meta-level language for describing semantic structures in some underlying/pre-existing object language (e.g., First Order Logic);
- among the aims of the developers of MRS has also been to get semantic representations for computational grammars of natural language which can be expressed in terms of feature structures

Introduction-Motivation (cont.)

- MRS was originally developed to accommodate the needs of the machine translation task in Verbmobil, where it has often been difficult or even unnecessary to resolve quantifier scope ambiguities;
- from this viewpoint, MRS has been very useful, since on the one hand offers the possibility to underspecify scope when it is unimportant, while on the other hand it ensures that scope is reconstructed when required;
- the smallest units of semantic (MRS) structure are the so-called *Elementary predications* (EPs), which correspond to a single relation (e.g., beyond(x,y));
- moreover, MRS provides for syntactically “flat” representations, since EPs are never embedded within one another;
- nevertheless, the necessary information about scope is preserved in the EPs, so that to be possible to get only the appropriate readings for a given sentence

Some technical details and definitions of MRS

- In order to achieve "flatness" in the represenatation, the structure of a conventional predicate calculus representation is simplified in certain ways:

(1) a) every big white horse sleeps

b) every(x, $\wedge$(big(x), $\wedge$(white(x), horse(x))), sleep(x))

c)

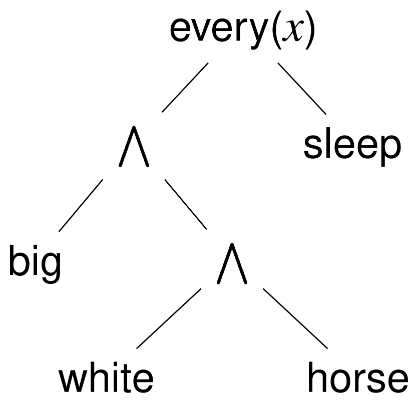

Figure 1: Tree representation of b)

Some technical details and definitions of MRS

- The modifications which serve to simplify the structure above are the following:
  - allowing $\bigwedge$ to be n-ary
  - ommiting the conjunction symbol (the order of the elements is not significant)
  - considering the nodes of the tree independently from any parent or daughter; this is achieved by giving each EP a special *handle* which identifies it as belonging to a particular tree node (*label*)

Some technical details and definitions of MRS

- (1) can then be represented as:

(2) h0: every(x)

```
    /   \
  h1     h2
```

h1: big(x), h1: white(x), h1: horse(x), h2: sleep(x)

- The fact that several EPs have the same handle (big, white and horse) can be explained since they are on the same node in the tree; thus, a simple flat list can be used:

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

Some technical details and definitions of MRS

DEFINITION 1 (Elementary predication (EP)):
An elemantary 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*

Example: $h2$:every(y, $h3$, $h4$)

Some technical details and definitions of MRS

DEFINITION 2 (MRS Structure):
An MRS structute is a tuple $<T, L, C>$ *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 constraint in C.*

Example:
(4) $<h0$, $\{h1$:every(x, $h2$, $h3$), $h2$:dog(x), $h4$:chase(x, y), $h5$:some(y, $h6$, $h7$), $h6$:white(y), $h6$:cat(y)$\}\{\}>$

$h0$ = top handle
$\{\}$ = empty bag of handle constraints

Constraints on scope relations

- with the help of scope constraints, restrictions on linkings of an MRS structure are described;
- of course, what is desirable is to be able to allow for underspecified quantifier restrictions in order to be able to get the different readings of given sentences;
- in which case, of course, the problem of unwanted scopes may arise

Constraints on scope relations

- the so-called *qeq* constraint (or $=_q$), which means equality modulo quantifiers, offers a solution in MRS to such problems by relating a handle in an argument position to a label;
- look at the following:

(5) a) every nephew of some fierce aunt runs

b) $<h1$, $\{h2$:every(x, $h3$, $h4$), $h5$:nephew(x, y), $h6$: some(y, $h7$, $h8$), $h7$:aunt(y), $h7$:fierce(y), $h10$:run(x)$\}$, $\{\}>$

c) $<h1$, $\{h2$:every(x, $h3$, $h4$), $h5$:nephew(x, y), $h6$: some(y, $h7$, $h8$), $h9$:aunt(y), $h9$:fierce(y), $h10$:run(x)$\}$, $\{$***h1*** $=_q$ ***h10, h7*** $=_q$ ***h9, h3*** $=_q$ ***h5*** $\}>$

- In b) the empty bag indicates that there are no handle constraints, whereas in c) one can see how the *qeq* constraints relate handles in argument position to the respective labels;
- thereby, it must be noticed that there is also a *qeq* relation between the top handle ($h1$) and the verb ($h10$);
- the main function of the *qeq* constraint is to write simple rules of semantic composition; thus, there is no need for special mechanisms which deal with quantifier storage

MRS in typed feature structures

Features:

- LTOP = local top; in a complete MRS, the top handle is qeq the local top ; the *ltop* serves to deal with floating and fixed scopal EPs
- LZT = list that introduces the bag of EPs
- HCONS = list of handle constraints
- SC-ARG = argument position handle (in *qeq* constraint)
- OUTSCPD = label handle

(6) a) $$\begin{bmatrix} \text{dog\_rel} & \\ \textsf{hndl} & \textsf{handle} \\ \textsf{inst} & \textsf{ref-ind} \end{bmatrix}$$

b) $$\begin{bmatrix} \text{every\_rel} & \\ \textsf{hndl} & \textsf{handle} \\ \textsf{bv} & \textsf{ref-ind} \\ \textsf{restr} & \textsf{handle} \\ \textsf{body} & \textsf{handle} \end{bmatrix}$$

c) $$\begin{bmatrix} \text{mrs} & \\ \textsf{top} & \boxed{1}\ \textsf{handle} \\ \textsf{ltop} & \boxed{7}\ \textsf{handle} \\ \textsf{lzt} & < \begin{bmatrix} \text{\_every\_rel} & \\ \textsf{hndl} & \boxed{2}\ \textsf{handle} \\ \textsf{bv} & \boxed{3}\ \textsf{ref-ind} \\ \textsf{restr} & \boxed{4}\ \textsf{handle} \\ \textsf{body} & \textsf{handle} \end{bmatrix}, \begin{bmatrix} \text{\_dog\_rel} & \\ \textsf{hndl} & \boxed{6}\ \textsf{handle} \\ \textsf{inst} & \boxed{3} \end{bmatrix}, \begin{bmatrix} \text{\_probably\_rel} & \\ \textsf{hndl} & \boxed{7} \\ \textsf{arg} & \boxed{8} \end{bmatrix}, \begin{bmatrix} \text{\_sleep\_rel} & \\ \textsf{hndl} & \boxed{9} \\ \textsf{arg1} & \boxed{3} \end{bmatrix} > \\ \textsf{h-cons} & < \begin{bmatrix} \text{qeq} & \\ \textsf{sc-arg} & \boxed{1} \\ \textsf{outscpd} & \boxed{7} \end{bmatrix}, \begin{bmatrix} \text{qeq} & \\ \textsf{sc-arg} & \boxed{4} \\ \textsf{outscpd} & \boxed{6} \end{bmatrix}, \begin{bmatrix} \text{qeq} & \\ \textsf{sc-arg} & \boxed{8} \\ \textsf{outscpd} & \boxed{9} \end{bmatrix} > \end{bmatrix}$$

Figure 2: Feature structure descriptions for *a) dog, b) every, c) every dog probably sleeps*

MRS in typed feature structures

- The non-feature-structure equivalent of (6) c) is shown in (7):

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

- it should be noticed that the handle numbers in (7) correspond to those in (6) c) except for the variable x which is represented as the coindexation $\boxed{3}$ in the feature structure; this feature structure form offers the possibility to integrate MRS with a feature-based grammar, e.g. one written within the framework of HPSG

References

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI Lecture Notes, Number 110. Standord: CSLI Publications.

Copestake, A., D. Flickinger, I. A. Sag, and C. J. Pollard (1999). Minimal Recursion Semantics: An Introduction. Ms., Stanford University.