Dependency semantics and composition

Ann Copestake

Computer Laboratory, University of Cambridge

September 5, 2017

Acknowledgments: DELPH-IN people, especially Emily Bender and Guy Emerson.

Outline.

Outline.

Computational compositional semantic representation

- Broad-coverage computational grammars.
- Any human language.
- Aim: capture all the semantically-relevant information in the syntax and inflectional morphology (plus productive derivational morphology).
- Underspecify distinctions that are not reflected in the syntax but are needed for well-formed representation.
- Parsing, realization, reasonable efficiency, statistical ranking, connection with lexical semantics …
- Work in LFG, TAG, CCG and other approaches but here DELPH-IN (HPSG or HPSGish).

DELPH-IN collaboration (`www.delph-in.net`)

- Hand-written English Resource Grammar (Flickinger 2000): about 80-90% coverage of ‘normal’ text.
- **NEW** Robustness (Packard and Flickinger, 2017).
- Other resource grammars: Jacy (Japanese), GG (German), SRG (Spanish), also varying size grammars for Norwegian, Portuguese, Korean, Chinese . . .
- tools for processing (Oepen, Packard, Callmeier, Carroll, Copestake et al), maxent parse/realization selection models (Redwoods Treebanks: Oepen et al 2002, etc)
- Shared semantic representations: Minimal Recursion Semantics (MRS: Copestake et al, 2005) and variants
- Grammar Matrix: Bender et al (2002).
- All Open Source since late 1990s.

A real example sentence

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

A real example sentence

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

modified quantifier

A real example sentence

Very few **of** the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

partitive

A real example sentence

Very few of the Chinese **construction companies** consulted were even remotely interested in entering into such an arrangement with a local partner.

compound nominal

A real example sentence

Very few of the Chinese construction companies **consulted** were even remotely interested in entering into such an arrangement with a local partner.

reduced relative

A real example sentence

Very few of the Chinese construction companies consulted were **even remotely** interested in entering into such an arrangement with a local partner.

modified modifier

A real example sentence

Very few of the Chinese construction companies consulted were even remotely interested in entering into **such an** arrangement with a local partner.

predeterminer

Some of the applications

- Email response (Flickinger, Oepen, et al: YY Technologies)
- Teaching English (Flickinger et al: EPGY, Redbird)
- Machine translation: e.g., Bond et al (2011)
- Information extraction and QA: e.g., MacKinlay et al (2009)
- Ontology extraction: e.g., Herbelot and Copestake (2006)
- Question generation: e.g., Yao et al (2012)
- Entailment recognition: e.g., Lien and Kouylekov (2014)
- Input for distributional semantics: e.g., Herbelot (2013)
- Detection scope of negation: e.g., Packard et al (2014)
- Robot control interface: e.g., Packard (2014)
- Logic to English (for teaching logic): Flickinger (2017)

This talk

1. Explain MRS (in a slightly different way from usual)
2. DMRS-v2: a variable-free representation that can represent scope. Interconvertible with ERG-MRS, and other semantic representation styles.
3. In progress work: doing composition directly in DMRS-v2.

This talk

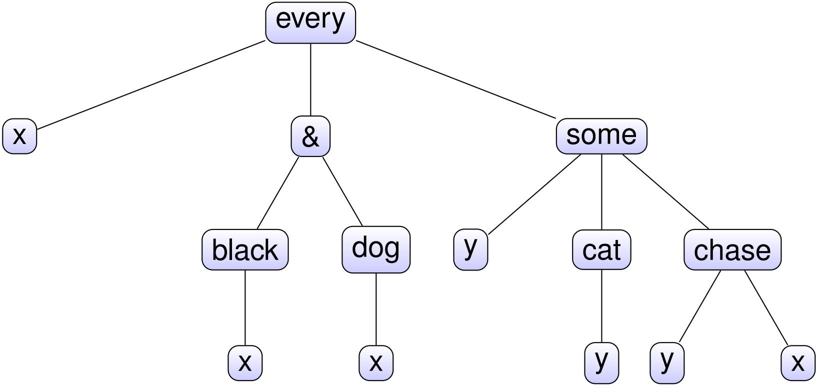
1. Explain MRS (in a slightly different way from usual)
2. DMRS-v2: a variable-free representation that can represent scope. Interconvertible with ERG-MRS, and other semantic representation styles.
3. In progress work: doing composition directly in DMRS-v2.

- Formalization in terms of graph structures, but concentrate here on intuitive explanation.
- Question for FSMNLP: could we usefully exploit finite-state methods?
- Question for linguists: what examples (English or otherwise) are interesting/challenging?

Outline.

Predicate calculus as a graph

every(x, black(x) & dog(x), some(y, cat(y), chase(y,x)))



Scope in graphs

every(x, black(x) & dog(x), some(y, cat(y), chase(y,x)))

- This is one reading of *some cat chased every black dog* (the other reading to be discussed shortly).
- For now, just interested in scopal relationships: a tree in most logical representation languages (variables later).
- Either use textual argument order (daughter order in trees) or explicit links (ARG1 etc).

Splitting up graphs

- Standard CS trick: convert graph to 'flat' structure by replacing links with identifiers.
  - every(x, black(x) & dog(x), some(y, cat(y), chase(y,x)))
  - l1:every(x,h1,h2), l2:&(h3,h4), l3:black(x), l4:dog(x), l5:some(y,h5,h6), l6:cat(y), l7:chase(y,x)
    h1=l2,h2=l5,h3=l3,h4=l4,h5=l6,h6=l7
- In MRS, connections via holes (h) and labels (l).
- Loukanova (2017): real variables vs 'memory locations' — holes and labels are memory locations.
- But, see later, status of 'real' variables?
- For those familiar with MRS: explicit conjunction for exposition now, but no event variables for this talk.

Underspecification (Hole semantics, MRS)

- Multiple graphs can be represented by a single flat structure with more complex constraints than equality.
- every(x, black(x) & dog(x), some(y, cat(y), chase(y,x)))
  some(y, cat(y), every(x, black(x) & dog(x), chase(y,x)))
- l1:every(x,h1,h2), l2:&(h3,h4), l3:black(x), l4:dog(x), l5:some(y,h5,h6), l6:cat(y), l7:chase(y,x)
  h1=l2,h3=l3,h4=l4,h5=l6, h2 and h6 left unspecified.
- If h2=l5 and h6= l7
  every(x, black(x) & dog(x), some(y, cat(y), chase(y,x)))
  If h6=l1 and h2=l7
  some(y, cat(y), every(x, black(x) & dog(x), chase(y,x)))
- But more complicated constraints needed in general.

MRS

- Use qeq constraints (equality modulo quantifiers) anywhere where scope is partially determined.
- Drop the explicit & and equate labels instead.
- l1:every(x,h1,h2), l2:black(x), l2:dog(x), l5:some(y,h5,h6), l6:cat(y), l7:chase(y,x)
  h1 qeq l2, h5 qeq l6
- Body of quantifier always unspecified.
- Quantifier outscopes all instances of its bound variable: left implicit in MRS.

Advantages of MRS 'flattening'

- Underspecify quantifier scope: record readings correctly but avoid exponential number of explicit readings. Simple types for NPs.
- Straightforward basic notion of compositionality: always accumulate 'elementary predications' and qeq constraints.
- Flat structure helpful for certain algorithms, including realization.
- MRS can be scoped (efficiently), and converted to other semantic representations (DRT etc), without further parsing or detailed lexical information.

MRS with explicit roles (cf feature structures)

l1:every(x,h1,h2), l2:black(x), l2:dog(x), l5:some(y,h5,h6), l6:cat(y), l7:chase(y,x) h1 qeq l2, h5 qeq l6

```
l1:every
   BV:  x
   RSTR: h1,
l2:black
   ARG1: x,
l2:dog
   ARG1: x,
h1 qeq l2,
```

```
l5:some
   BV:  y
   RSTR: h5,
l6:cat
   ARG1: y,
h5 qeq l6,
l7:chase
   ARG1: y
   ARG2: x,
```

- Conversion to argument names requires general conventions (no detailed thematic roles).
- Generalize between ARG1, ARG2 (in RMRS).

MRS in feature structures

```
[ LBL: hndl <1>
  PRED: every
  BV: ind <2>
  RSTR: hndl <3> ],
[ LBL: hndl <4>
  PRED: black
  ARG1: <2> ],
[ LBL: <4>
  PRED: dog
  ARG1: <2> ],
[ LBL: <5>
  PRED: some
  BV: ind <6>
  RSTR: hndl <7>],
```

```
[ LBL: <8>
  PRED: cat
  ARG1: <6>],
[ LBL: <9>
  PRED: chase
  ARG1: <6>
  ARG2: <2>],
   [ qeq
     HOLE: <3>
     LABEL: <4>],
   [ qeq
     HOLE: <7>
     LABEL: <8>]
```

MRS in feature structures

- Encoding via a directed acyclic graph, EPs in a list.
- Things in lowercase (types) may be in a hierarchy, things in capitals (features) cannot.
- Lots of different ways of encoding, standardized for DELPH-IN Matrix grammars, simplified here.
- Main point here: coindexation/reentrancy (shown by <1> etc) instead of variables. i.e., links.
- Hence: ‘real’ variables are ‘memory locations’.
- Conversion to standard representation relies on assumption that anything not linked together is distinct (cf equality between conventional variables).

MRS: some issues

MRS is (very) useful, but:

- Very difficult to explain/read MRS as used in ERG (ERS). Not an easy target for machine learning approaches.
- Composition constraints: algebra only partially successful.
- Variables are not doing much (memory locations), and complicate algorithms.
- MRS support within DELPH-IN has become tuned to ERG specifics.
- Predicate modifiers.
- One solution: DMRS (DMRS-v2).

ERG MRS

```
TOP     h0
INDEX   e2
RELS    < [ _some_q_indiv⟨0:4⟩
            LBL   h4
            ARG0  x3
            RSTR  h5
            BODY  h6 ],
          [ _cat_n_1⟨5:8⟩
            LBL   h7
            ARG0  x3 ],
          [ _chase_v_1⟨9:15⟩
            LBL   h1
            ARG0  e2
            ARG1  x3
            ARG2  x8 ],
          [ _every_q⟨16:21⟩
            LBL   h9
            ARG0  x8
            RSTR  h10
            BODY  h11 ],
          [ _black_a_1⟨22:27⟩
            LBL   h12
            ARG0  e13
            ARG1  x8 ],
          [ _dog_n_1⟨28:32⟩
            LBL   h12
            ARG0  x8 ] >
HCONS   < [ qeq
            HARG  h5
            LARG  h7 ],
          [ qeq
            HARG  h0
            LARG  h1 ],
          [ qeq
            HARG  h10
            LARG  h12 ] >
```

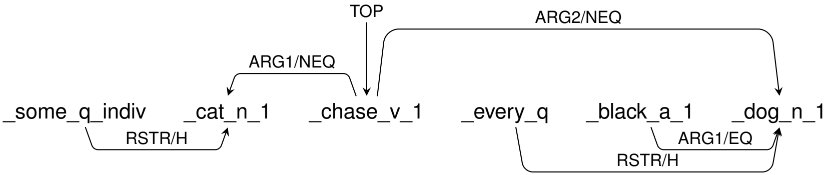
ERG MRS: things I’m not mentioning . . .

- predicate names for words are of the form _chase_v_1
- for constructions, no leading underscore
- character positions are recorded
- events and ‘event’s (more soon)
- tense, aspect, plurality etc: recorded as attributes of variables
- information structure, anaphora

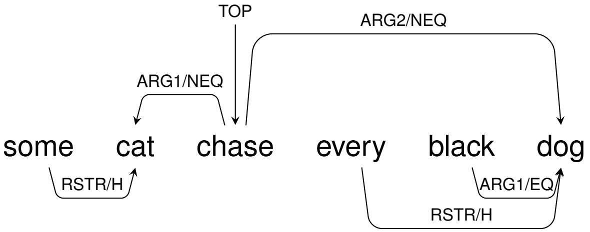
Demos

Michael Goodman
http://chimpanzee.ling.washington.edu/demophin
Ned Letcher
http://delph-in.github.io/delphin-viz/demo/
Woodley Packard: ACE parser/generator

ERG DMRS


TOP
ARG2/NEQ
ARG1/NEQ
_some_q_indiv
_cat_n_1
_chase_v_1
_every_q
_black_a_1
_dog_n_1
RSTR/H
ARG1/EQ
RSTR/H

DMRS notation for this talk

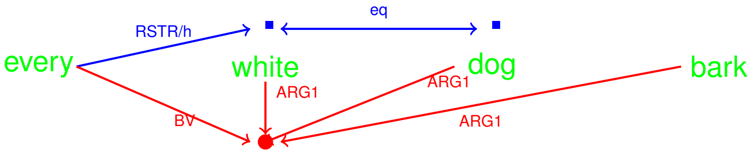


Outline.

Getting rid of variables

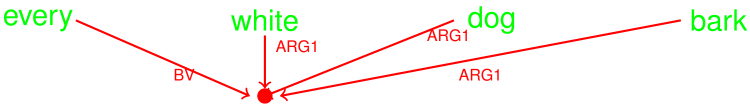
FS encoding shows we can use a graphical representation and don't need variables as such (at least for composition).

- every white dog barks
  every(x, white(x) & dog(x), bark(x))
  every(x,h1,), l1:white(x), l1:dog(x), bark(x), h1 qeq l1



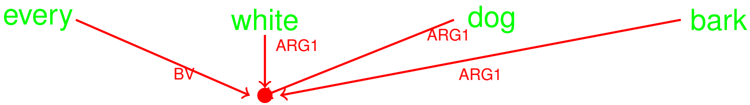
Do we need all these nodes? Why not link predicates directly? i.e., can we use semantic dependencies?

Getting rid of variables: the redundant link problem



- Remove nodes corresponding to variables, capture semantics by links between predicates.
- But lots of links:
  every to white
  every to dog
  every to bark
  white to dog
  white to bark
  dog to bark

Getting rid of variables: deciding on links

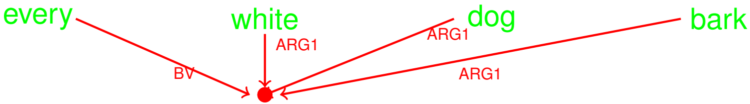


Given a semantic relationship between two or more entities, captured by variables in predicate calculus, need to decide:

- which entities to link (if more than two share a variable)
- direction of the link
- whether/how to combine links with same source-target (relevant for DMRS because of links representing scope).

Canonical linking

Intuitively:



should be:

But need general motivation, which works thoughout the grammar for every language and without using details of syntax.

Canonical linking: first attempt

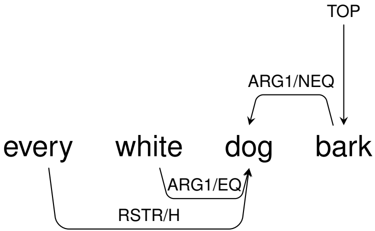
- Canonical linking via additional variables:
  - MRS as used in ERG: almost every predicate is associated with its own variable:
    every big dog barks loudly
    Fully scoped form:
    every(x, big(e1,x) & black(e2,x) & dog(x), bark(e3,x) & loud(e4,e3))
  - This allows a canonical link between predicates: each link points to the predicate 'owning' the variable.
- Oepen uses this property for EDS (additional events were partly introduced for this reason).
- Also first Dependency MRS (Copestake 2009).
- But requires lots of 'events', with limited justification.

Canonical linking: functor-argument relationships

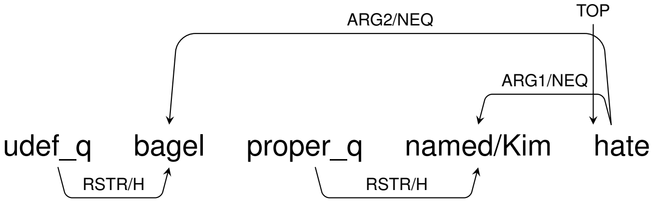
Observation: HPSG was partly inspired by categorial grammar.

- Functor-argument relationship for syntax/semantics: COMP, SUBJ, MOD etc are slots to be instantiated.
- Functor is usually the HEAD, except for modifier constructions, and determiners, where two-way selection.
- Hence canonical representation for semantic dependency links (though semantics doesn't always follow syntax).
- Representation not dependent on approach to events, based on underlying HPSG principles, should be adaptable for other frameworks.
  Additional events give back-door access . . .
- DMRS-v2: looks almost exactly like original DMRS (but undirected EQ links in DMRS-v1 are directed in DMRS-v2).

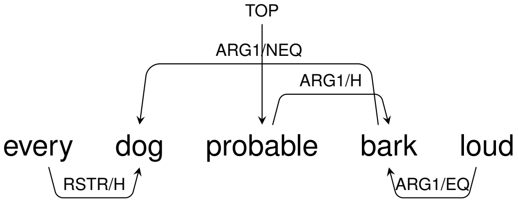
every white dog barks

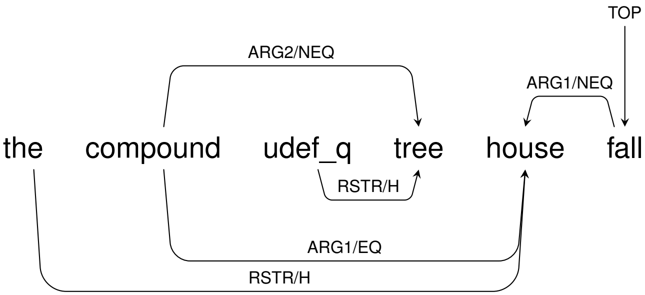


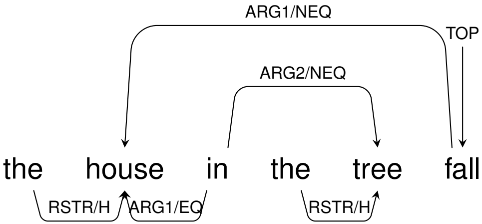
bagels, Kim hates



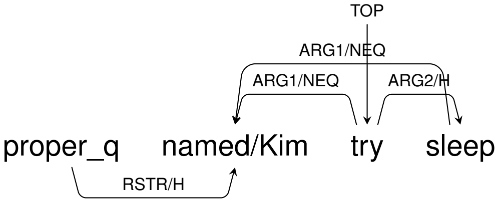
every dog probably barks loudly



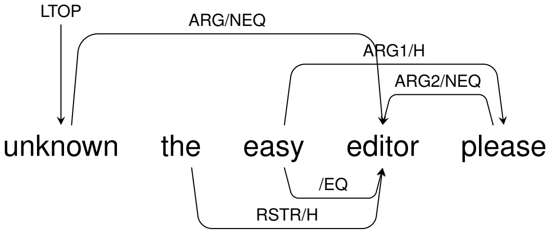

TOP
ARG2/NEQ
ARG1/NEQ
the
compound
udef_q
tree
house
fall
RSTR/H
ARG1/EQ
RSTR/H


ARG1/NEQ
TOP
ARG2/NEQ
the
house
in
the
tree
fall
RSTR/H
ARG1/EQ
RSTR/H

Kim tries to sleep



the easy editor to please



DMRS flexibility (maybe . . .)

- Link labels can be underspecified (as in RMRS).
- Scopal vs non-scopal modifier: link between modifier and modifiee underspecified in its scopal component.
- Predicate modifiers (possibly MRS as well).
- Non-tree scopal structures:

  We could and should talk.
- PP-attachment: no crossing condition allows derivation of all possible attachments (at least in simple cases).

Warning: none of this demonstrated on any scale!

DMRS-v2 (in progress)

- Same possibility of conversion from ERG-MRS to DMRS.
- Non-trivial grammars using DMRS directly have been created.
- Possible DMRS alternative for the Matrix (Emerson, Bender).
- Natural approach to composition (last part of talk).
- DMRS scoping: much like MRS.

Outline.

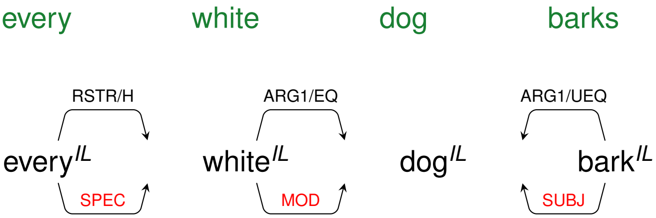
Compositionality and broad-coverage grammars

- Underlying intuition: semantics should ‘mirror’ syntax, but difficult to achieve in a large-scale grammar.
- Grammar engineering perspective: capture generalizations, limit ad hoc aspects of grammar. Also realization and scopability of *MRS.
- Learnability (human and machine).
- Traditionally, HPSG has allowed great flexibility in syntax-semantics relationship.
- MRS algebra (Copestake et al, 2001; Copestake (2007): tried to constrain composition, but not fully successful.
- Discussion of MRS compositionality (and contrast with AMR) in Bender et al (2015: IWCS).

Compositionality in DMRS

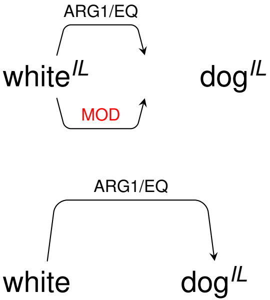
- Intuition: extract syntactic dependencies from an HPSG, look at exceptions to isomorphism with DMRS.
- Intuition: lexical exceptions OK (multiword expressions).
- Model what is actually done in HPSG/DELPH-IN/Matrix in semi-formal DMRS/dependency notation, and then see what constraints could be feasible.
- Abstract away from details of the feature structure grammars.
- Follow original algebra in limiting access to *MRS: LTOP (scope), INDEX (individuals) and XARG.

Stage 1: initialize elements



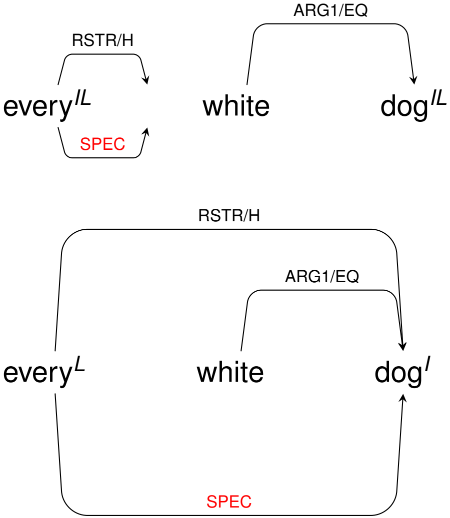
Complexities: lexemes with null semantics or complex semantics; construction predicates; multi-word expressions.

Stage 2: white dog



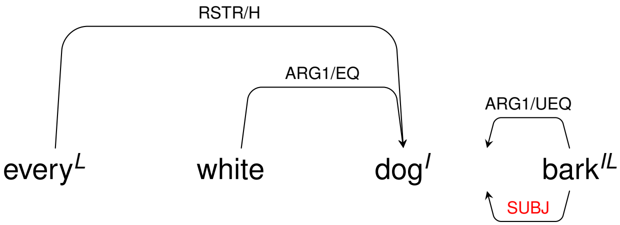
- I = INDEX, L = LTOP
- EQ, NEQ and UEQ links select INDEX
- INDEX of phrase comes from HEAD
- LTOP comes from HEAD (except for scopal modifiers etc)
- syntax links dropped when saturated

Stage 2: every white dog



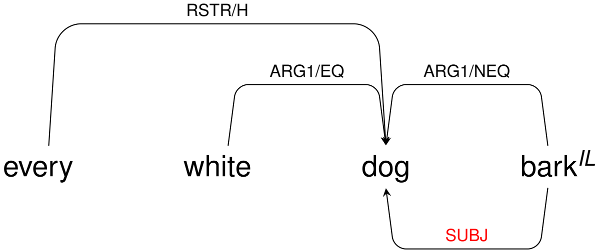
- Only semantically relevant selection is SPEC.
- LTOP on quantifiers is a choice point.

Stage 2: every white dog barks



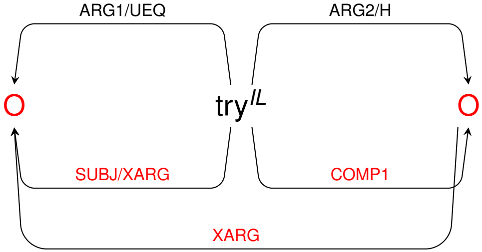
- UEQ on ARG1 from verb because could be in a relative clause.

Stage 2: every white dog barks

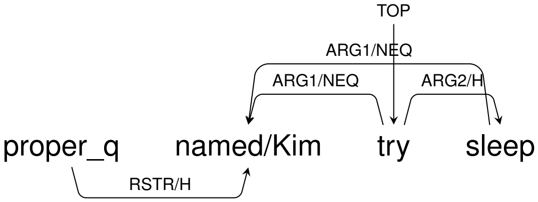


- UEQ specialised to NEQ
- Restrictive relative (dog which sleeps) would be ARG1/EQ

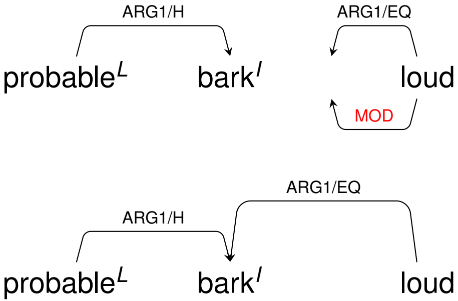
XARG



XARG: Kim tries to sleep



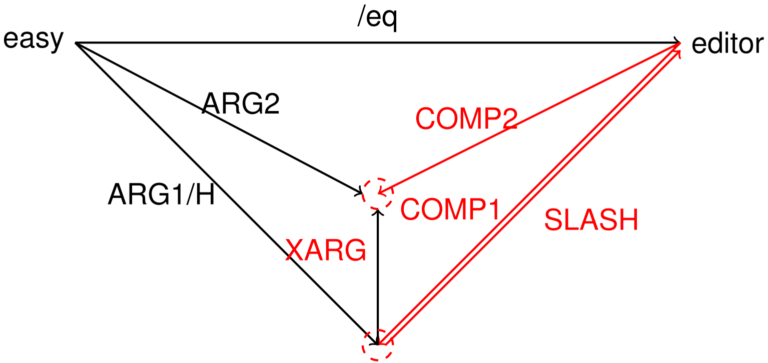
probably barks loudly



DMRS (unlike MRS) allows:
probable(bark(e,x) & loud(e)) from
((probably barks) loudly)
Not so interesting for English but relevant for other languages.

easy editor isn't easy ...

- Analysis based on Flickinger and Nerbonne (1992).
- Makes use of the transferrable subcat principle.
- May not want to allow this!



Constraints

- Current status: trying to work out best notation and putative constraints before implementation.
- Plan is to work out consequences with smaller grammars and (eventually, maybe) do a native DMRS version of the ERG.
- May not be ‘nice’ constraints:
  - Constraints of the form ‘no more than four’.
  - Possible that constraints are (partly) language-specific.
  - Violations might be statistical: not that something never happens, but that it is rare.
- Incremental (strictly left-to-right) DMRS composition looks possible but raises additional challenges.

Outline.

Conclusions

- Introduced MRS, DMRS-v2, DMRS composition.
- Emphasis of the current work is on doing things with large-scale resources: empirical investigation combined with theoretical investigation.
- Composition constraints at an early stage.
- Questions:

Conclusions

- Introduced MRS, DMRS-v2, DMRS composition.
- Emphasis of the current work is on doing things with large-scale resources: empirical investigation combined with theoretical investigation.
- Composition constraints at an early stage.
- Questions:
  - Could we usefully exploit finite-state methods?
  - What examples (English or otherwise) might be interesting/challenging?