Contents

1 Introduction

1.1 For some tasks (e.g., generation, transfer), MRSs must conform to the semantic model to be useful

Consider: `http://delph-in.github.io/delphin-viz/demo/#input=Abrams%20knew%20that%20it%20rained.&count=5&grammar=ergtrunk-uw&mrs=true`

1.2 The model involves several structures, including:

- variable hierarchy

```
e < i : SF sf, TENSE tense, MOOD mood, PROG bool, PERF bool.
```

- role inventory

```
ARG0 : i.
```

- property hierarchy

```
past < tensed.
```

- predicate hierarchy

```
_abaft_p < mod.
```

- predicate synopses:

```
_abaft_p : ARG0 e, ARG1 x { NUM sg }.
```

1.3 These are encoded in the SEM-I, which is automatically generated from a grammar

- That is, the SEM-I is a product of a grammar (hence "interface" and not "model", I guess)
- Automatic generation is not bug-free

```
_abbreviate_v_1 : ARG0 e, ARG1 i, ARG2 i, [ ARG3 i ].
```

2 Semi-Useful Models

2.1 SEM-I (2016) was a great improvement

Used in PyDelphin (IndexedMRS) and in my dissertation (generating with the ERG)

2.2 Remaining issues (taken from `http://moin.delph-in.net/SemiRfc`)

- Associating related predicates (e.g., senses of "eat", mass/count) :fcb:
- Making computed hierarchy visible :aac:
- Improve or remove argument optionality marking :oe:
- Use SEM-I to encode order for roles, morphosemantic properties :mwg:
- Encode semantic effects of phenomena like control :mwg:
- Encode HCONS and ICONS relations (just the relation hierarchy) :mwg:
- Add CARG to relevant predicate synopses :mwg:
- Group predicates by common synopsis rather than repeating synopses :mwg:

3 Uses of underspecified variables

See: `https://delphinqa.ling.washington.edu/t/new-uses-of-underspecified-variables-in-th`

```
u := *top*.                  u
i := u.                     / \
p := u.                    i   p
e := i.                   / \ / \
x := i & p.              e   x   h
h := p.
```

3.1 Observed uses

- `i` as intrinsic variables of number constructions in the ERG
- `i` as intrinsic variables of scopal modifiers (ERG 2018, but also for `neg`, `addressee`, etc. in 1214)
- `i`, `p`, `u` for dropped arguments

3.2 Proposal for updated variable hierarchy

```
u := *top*.                    u
i := u.                       /|\
d := u.                      i d p
p := u.                     / \|/ \
e := i.                    e   x   h
x := i & d & p.
h := p.
```

4 Fixed Arity

Required for conversion to logical forms.

Where a predicate has exactly **n** regular arguments and **m** scopal arguments.

4.1 All non-scopal arguments occur before scopal arguments?

```
_advocate_v_1 : ARG0 e, ARG1 i, ARG2 h, ARG3 i.

                ^ ^
```

4.2 All synopses have compatible types?

```
_affect_v_1 : ARG0 e, ARG1 u, [ ARG2 h ].
_affect_v_1 : ARG0 e, ARG1 i, ARG2 i.
```

4.3 All synopses have the same number of arguments?

```
_advise_v_1 : ARG0 e, ARG1 i, ARG2 i, [ ARG3 h ].
_advise_v_1 : ARG0 e, ARG1 i, ARG2 p, ARG3 h.
_advise_v_1 : ARG0 e, ARG1 i, ARG2 h.
```

4.4 What about morphosemantic properties?

```
_afternoon_n_of : ARG0 x { GEND n, NUM sg }.
_afternoon_n_of : ARG0 x { IND + }.
_afternoon_n_of : ARG0 x { NUM pl }.
_afternoon_n_of : ARG0 x { NUM pl, IND + }, ARG1 u.
_afternoon_n_of : ARG0 x.
_afternoon_n_of : ARG0 x { GEND n, NUM sg, IND + }, ARG1 u.
```

5 When is fixed arity useful?

5.1 Conversion to logical form (or IndexedMRS)

```
$ delphin convert --to indexedmrs --sem-i ~/grammars/erg-trunk/etc/erg.smi the-chef.mr
 < h0, e2:PROP:PAST:INDICATIVE:-:-,
   { h4:_the_q<0:3>(x3:3:SG:GENDER:+:PT, h5, h6),
     h7:_chef_n_1<4:8>(x3),
     h8:def_explicit_q<9:14>(x9:3:SG:GENDER:BOOL:PT, h10, h11),
     h12:poss<9:14>(e13:PROP:UNTENSED:INDICATIVE:-:-, x9, x3),
     h12:_soup_n_1<15:19>(x9),
     h7:_spill_v_1<24:31>(e14:PROP:PAST:INDICATIVE:-:-, i15, x9),
     h1:_quit_v_1<32:37>(e2, x3, i16) },
   { h0 qeq h1,
     h5 qeq h7,
     h10 qeq h12 },
   { e14 topic x9 } >
```

5.2 MRS validation

Is an MRS valid if there exist any predicates whose synopses subsume their instances in the MRS?

5.3 DMRS composition

E.g., link the top of subgraph **q** as the ARG1 starting from some node **n**..

- Need to know if **n** can scopally take ARG1.
- What if multiple synopses exist... cannot commit too early

5.4 Unexpressed Arguments in DMRS

Use special `__unexpr__` symbol for unexpressed nodes where MRS uses an underspecified variable.

- Apples were picked. `http://delph-in.github.io/delphin-viz/demo/#input=Apples%20were%20picked.&count=5&grammar=ergtrunk-uw&mrs=true&dmrs=true`

  Insert `__unexpr__` node for ARG1 of `_pick_v_1`, link with ARG1/NEQ (?)

- Baking is easy. http://delph-in.github.io/delphin-viz/demo/#input=Baking%20is%20easy.&count=1&grammar=erg-uw&mrs=true&dmrs=true

  Just insert __unexpr__ nodes for ARG1 of _bake_v_1 and ARG2 of _easy_a_for, link with ...

- Kim believed. http://delph-in.github.io/delphin-viz/demo/#input=Kim%20believed.&count=1&grammar=erg-uw&mrs=true&dmrs=true

  ARG2 of believe is p-variable. Do we link ARG2/H, ARG2/HEQ, ARG2/NEQ to __unexpr__?

  ```
  _believe_v_1 : ARG0 e, ARG1 i, ARG2 h.
  _believe_v_1 : ARG0 e, ARG1 i, ARG2 i.
  ```

  If arity is fixed, we could narrow it down:

  ```
  _believe_v_1 : ARG0 e, ARG1 i, ARG2 h.
  _believe_v_2 : ARG0 e, ARG1 i, ARG2 i.
  ```