1   (Closed) Clausal Complements

In AMR, we usually use a very simplistic view of semantic headedness, so that complementation is usually a link between the main/matrix verb and the "head" of the embedded clause, as in[1]:

- They forgot that they had voted.
  ```
  (f / forget-01
        :ARG0 (t / they)
        :ARG1 (v / vote-01
               :ARG0 t))
  ```

This conflates reference to the single concpet (the voting event), the concept in context (the voting event, with "they" as a subject), and the truth value and related features of the verb. One might argue that the head of "that they had voted" is the fact that it is actually happened in the past, something we don't currently care about at all.

But we do have a model for handling sentences like [105-109], where one might argue that the "thing causing surprisal" is the relationship between "early" and the event itself. [2] In AMR, our approach is to simply "reify" the temporal relation between the two, assigning the relationship a variable and arguments:

```
(1)   (s / surprise-01
          :ARG1 (i / i)
          :ARG0 (b / be-temporally-at-91
               :ARG1 (v / vote-01
                       :ARG0 (t / they))
               :ARG2 (e / early)))
      "That they voted early surprised me."
```

This can be elegant. We don't currently do this kind of model for more core arguments, but if we extended the premise to them it would look like this:

```
(2)   (d / deny-01
```

---

[1] For those unfamiliar with reading AMRs: AMRs are directed acyclic graphs with numbered arguments. So "he likes talking", which would be "(l / like-01 :ARG0 (h / he) :ARG1 (t / talk-01 :ARG0 h))", wherein nodes such as "l / like-01" have a variable "l" that can be referenced later in the sentence, and a sense ("like-01") that specifies the exact meaning of the concept used (see http://verbs.colorado.edu/propbank/*framesets-english*/like-v.html). The sense of "like" used also tells the reader how to interpret the ARG0, ARG1, etc., although there is a a general pattern that "ARG0" means the proto-agent and "ARG1" the proto-patient or undergoer. The AMRs are presented in such a way that the dependent node is to the right of a given argument, and that the head is up and to the left, so that it behaves like a tree. The inverted relations (when one sees an "ARG#-of"), in turn, simply mean that it means that the higher element in the "tree" (as we represent these as trees) is the dependent argument of the relationship, so that "(l / like-01 :ARG0 (h / he) :ARG1-of (t / talk-01 :ARG0 h))" would be closer to "he talked about liking".

[2] For "early", one might say that early actually is marking the temporal relationship between two things, but I don't think one can make the same analysis for "That they voted Tuesday surprised me".

```
        :ARG0 p
        :ARG1 (b / be-arg0-of-91
              :ARG1 (p / person :name (n / name :op1 "John"))
              :ARG2 (s / steal-01
                    :ARG1 (y / yogurt))))
   "John denied being the one that stole the yogurt"
```

Instead, AMR treats such examples more shallowly as:

```
(3)  (d / deny-01
        :ARG0 p
        :ARG1 (s / steal-01
              :ARG0 (p / person :name (n / name :op1 "John"))
              :ARG1 (y / yogurt)))
   "John denied being the one that stole the yogurt"
```

I'd love to see if other representational models have more elegant solutions to this kind of issue.

2 Coordination

On the multiplicity of events

How many events to actually "represent" is always a brutal topic, at least for AMR. On one level, this is easy in AMR: if a given verb implies many different versions of itself (as in "the dog barked Monday and Wednesday"), then multiple instances of that verb, each with its own unique variable, can be introduced. This means that the more complicated issues of coordinated arguments can often be dealt with by essentially having a copy of the predicate, as in:

```
(4)  (a / and
        :op1 (b / bark-01
              :ARG0 (d / dog)
              :time (d2 / date-entity
                    :weekday (m / monday)))
        :op2 (b2 / bark-01
              :ARG0 d
              :time (d3 / date-entity
                    :weekday (w / wednesday))))
   The dog barked on Monday and on Wednesday.
```

This means that annotators (and thus systems/grammars generating AMRs) would have to distinguish between distributed readings of coordinated nouns, and collective readings:

```
(5)  (d / dance-01
        :ARG0 (a / and
              :op1 (p / person :name (n / name :op1 "Abrams"))
```

```
                          :op2 (p2 / person :name (n2 / name :op1 "Browne"))))
          Abrams and Browne danced (with each other)

(6)   (a2 / and
          :op1 (d3 / dance-01
                  :ARG0 (p / person :name (n / name :op1 "Gene" :op2 "Kelley")))
          :op2 (d2 / dance-01
                  :ARG0 (p2 / person :name (n2 / name :op1 "Justin" :op2 "Bieber"))))
          Gene Kelley and Justin Bieber danced (in two very different dancing events)
```

This "seems to work"; annotators do split things up when necessary. It also means that we handle things like "respective" constructions in a relatively sane way:

```
(7)   (a2 / and
          :op1 (e / eat-01
                  :ARG0 (p / person :name (n / name :op1 "John")))
          :op2 (d2 / drink-01
                  :ARG0 (p2 / person :name (n2 / name :op1 "Mary"))))
          John and Mary ate and drank, respectively.
```

Coordination and Multiword Expressions

The "hem and haw" example gets to something that is a little "strange" in AMR; because of the lexical definition of roles (everything is "Arg0", etc.), we decompose an participant's role in relation to something like "hem and haw" into them being an argument of a "hem" predicate and a separate "haw" predicate. This is usually fine idea; "John sang and danced" links up the variable for "John" separately to sing-01 and to dance-01.

This means that in AMR, we have two options for hemming and hawing. One must either define such MWEs in the lexicon as a single concept (which we do not do right now, but may someday want to implement), or would do this old trick of having a sense of "hem" that means the "the 'hem' of 'hemming and hawing", and a sense of "haw" that is specific for "hemming and hawing", as in:

```
(8)   (a / and
          :op1 (h / hem-02
                  :ARG0 (h3 / he))
          :op2 (h2 / haw-01
                  :ARG0 h3))
          He hemmed and hawed.
```

3   Ellipsis

In AMR, since we are asking people to just do the "sentence meaning", ellipsis is sometimes not a problem. For all of the "ellipsis" examples, if the elided event was mentioned in the same sentence, annotators would be supposed to simply

add the variable back in; we make no distinction between syntactic or surface anaphora and pragmatic/deep reference.

One question, then, is what we do when one cannot make the reference link, either due to only doing in-sentence coreference (a practical constraint we would hope to eliminate) or because of lack of information. Our model is to just assume these to be a generic "event" structure[3]:

```
(9)   (e / event-01 :mode interrogative
            :ARG0 (y / you))
      Did you?
```

4   Identity Copula

AMR, as a rule, removes all copulae from the semantic representation: everything is some form of more specific predication, as in:

```
(10) (h2 / have-org-role-91
            :ARG0 (p3 / person :name (n2 / name :op1 "Browne"))
            :ARG2 (p / person
                  :ARG0-of (m / manage-01)))
      Browne is a manager.
```

```
(11) Abrams is Browne.
      (p / person :name (n / name :op1 "Abrams") :name (n2 / name :op1 "Browne"))
```

Thus when "identity copula" constructions are complex ways of doing nominal relations, our solution is that we simply have predicates for nominal relations, and are allowed get them. Presumably a grammar would have to learn from the words; "Manager" would have to entail having the organizational role of "person who manages", for example.

The more interesting question from this perspective is terms like "the plan is to sleep more". We would ignore some of the referential complexities here and encode it the same way you'd encode "sleeping more is planned", perhaps losing the additional referentiality of "the plan" being nominal:

```
(12) (p / plan-01
            :ARG1 (s / sleep-01
                  :extent (m / more)))
```

I think that if one had a model in which coreference was done in separate links (instead of using variable binding), then I presume they could model is as asserting identity between "the thing planned" and sleeping more". This has advantages and disadvantages if one then ran into a sentence like "then we changed the plan to just hanging out.", where one might have to then do some changing of the identity links.

---

[3]This is not yet "on the books"; we use "event" without an argument structure, but have talked about implementing "event-01" so as to handle examples like these.

5 Nominalization

As AMR annotators are allowed to introduce dummy elements (from a defined type inventory), free relatives like "what the dog chased bothered browne" get treated by introducing a referent for the filer. Thus "the thing that the dog chased" and "what the dog chased" are, for us, identical:

- ```
  (b / bother-01
        :ARG0 (t / thing
                  :ARG1-of (c / chase-01
                              :ARG0 (d / dog)))
        :ARG1 (p / person :name (n / name :op1 "Browne")))
  ```

One big question with nominalizations is the same one as "plan" before; I often find it theoretically hard to discern reference to a core predication from reference to its inherent results or outcomes. If you say "The student evaluations were negative", we are encoding "evaluation" as being identical to "evaluate":

```
(13) (n / negative
          :ARG2-of (e / evaluate-01
                    :ARG0 (p / person
                              :ARG0-of (s / study-01))))
     The student's evaluations were negative (the "judgement" role of the evaluate event is
```

In many ways, "evaluation" is an event in which one evaluates. Yet in other ways, one might postulate that "evaluation" also introduces the results of that evaluating event as a discourse participant, and that those results can be argued about or re-examined. Thus the more precise AMR approach would be to do this as:

```
(14) (n2 / negative
          :domain (t / thing
                    :ARG2-of (e / evaluate-01
                          :ARG0 (p / person
                                    :ARG0-of (s / study-01)))))
     The student's evaluations were negative
```

6 Comparatives

AMR is often "sloppy" about comparatives, marking the shallow reading of "the dog is more old than the cat"

```
(o / old
     :domain (d / dog)
     :degree (m / more
           :compared-to (o2 / old
                 :domain (c / cat))))
```

That said, we can theoretically do the full treatment of showing the scale of comparison, as in:

```
(o / old
     :domain (d / dog)
     :degree (m / more
            :compared-to (o2 / old
                   :domain (c / cat))))
```

We therefore want to just express "the dog is more aggressive than the cat is" by encoding the "aggressiveness" properties of both animals and marking a comparison between them. This runs into the problem that we have no model for referencing a property of an object without asserting the default value for it: talking about the agressiveness of the dog, in AMR, is the same as asserting that the dog is aggressive. We suspect that one does not always want "The dog is more aggressive than the cat" to entail that the cat is aggressive.

The other question is what to do when "the thing compared with" is not another referent or an unstated null instantiation, but rather a prior version of the same argument. We currently don't introduce the "comparand" when it's unmentioned anyways, as in:

(15) (m / move-01
           :ARG1 (i / it)
           :manner (s / slow
                  :degree (m2 / more)))

However, imagine that this above sentence was in the context of:

(16) "My dog wasn't as energetic as it once was. It complained more. It slept more. It moved more slowly."

I suspect that we would theoretically want "moved more slowly" to related to a prior motion event ("It moved more slowly (than it used to move)"). I'm not sure if we'd ever model that in AMR, and would love to see if other models have better treatments.

7   Control Relations

Since AMR, again, simply encodes relations between every mentioned participant and all of the semantic roles they participate in, control is pretty easy for us ("us" being those doing the annotation; this is a more complex issue for AMR parsing.)

The one theoretical issue that does come up (which I presume other models are handling better than we are) is the question of whether one wants to track control relations over unstated arguments. In AMR, "I want to be someone impressive to meet" does not assert the unstated "person being impressed"; pursuantly, the fact that they are also the agent of "to meet" is completely lost. I presume that other models can handle this with unstated arguments.

8 Measure Phrases

AMR does normalization of measurement phrases using an inventory of measurement units, as in:

```
(17)  (a2 / age-01
          :ARG1 (d / dog)
          :ARG2 (t / temporal-quantity :quant 10.5
                   :unit (y / year)))
      The dog was ten-and-a-half years old.
```

This seems like a good solution, although the more specific subfields one works in, the more vast such an inventory of quantity entities becomes.

9 Parentheticals

We don't actually touch "background" vs "foreground" distinctions, nor other information-structural issues. If one is looking at the difference between "An old friend (Abrams) arrived" and "Abrams (an old friend) arrived", we would, in fact, treat them identically. I do think these have different entailments (the first implies somewhat that the audience knows who Abrams is), but we necessarily abandon all argument realization information like this.

10 Relative clauses:

One main thing about the AMR relationship to relative clauses is that we do not distinguish between types of relative clauses (or mark them at all), nor the information structure that they tend to encode, which means that we treat alternations like these as identical:

(18) He had passed the information to a friend, who sent it along to the New York Times.

(19) He had passed the information to the friend who had sent it to the New York Times.

However, the relative clauses I find intriguing are the ones with reference to events and discourse units, as in:

(20) It rained yesterday, **which** troubled me.

(21) He's both a lawyer and a lexicographer (**which** seems a bit like being both a narcotics dealer and a DEA agent).

These are doable in AMR, as we treat relative claues as just being the semantic relations involved, but I'd be interested to see how other models handle them:

```
(22)  (t / trouble-01
          :ARG0 (r / rain-01
                  :time (y / yesterday))
          :ARG1 (i / i))
      It rained yesterday, which troubled me.
```

This relates to full "discourse level" relative clauses, which are probably not much of a problem unless one deals with proper conversation, as in:

(23) Chairs have been known to be pulled out for laughs; knees have buckled, as have lower backs. But, generally speaking, once you're seated, you're pretty much out of the woods. **All of which is to say** that if elected, I will work tirelessly to defend your right to sit, and vow to fight with all my powers the vast but unacknowledged anti-sitting lobby that works to dominate and destroy the fabric of our American life.

Now one should probably call this a construction, and say that it's anaphora, but you will still need to get hierarchical structure over discourse to resolve it. Our related discourse model (RED, which might be done over AMR) or something like RST might be able to lump together those units and refer to the container, but it's hard to figure out how to have that integrated into a semantic representation.