November 25: More on dynamics

1 Reviewing the basic picture

1.1 Example

We'll begin by reviewing a basic example of discourse anaphora relative to a simple model. Here is a simple two-sentence text, where there is anaphora from the second sentence into the first:

(1) Bill_3 left. He_3 was tired.

We typed sentences as *relations* on assignment functions, type $\pi ::= \langle a, \langle a, t \rangle\rangle$, to allow for two things:

1. That processing a sentence could have an effect on the discourse referents (drefs) available to interlocutors.
2. That this process could fail (i.e. denote an *empty* relation) when a sentence communicates something false.

Recall now our semantics for the individual sentences (type π):

$$[\![\text{Bill}_3 \text{ left}]\!] = \lambda g. \begin{cases} \{g^{[\text{b}/3]}\} \text{ if } \textit{left}'(\text{b}) \\ \text{else } \emptyset \end{cases}$$

$$[\![\text{he}_3 \text{ was tired}]\!] = \lambda g. \begin{cases} \{g\} \text{ if } \textit{tired}'(g(3)) \\ \text{else } \emptyset \end{cases}$$

...and our lexical entry for conjunction (type $\langle \pi, \langle \pi, \pi \rangle\rangle$), which *composes* the two relations denoted by the individual conjuncts while giving precedence to the left conjunct:

$$[\![\text{and}]\!] = \lambda r.\lambda l.\lambda g.\{h \mid k \in l(g) \text{ and } h \in r(k)\}$$

Put these pieces together, we have the following as the meaning for the text as a whole. Relational composition (the semantics of conjunction) feeds the assignments as updated by the first conjunct to the second conjunct:

$$\lambda g.\{h : k \in [\![\text{Bill}_3 \text{ left}]\!](g) \text{ and } h \in [\![\text{he}_3 \text{ was tired}]\!](k)\}$$

Assuming Bill left, in which case $[\![\text{Bill}_3 \text{ left}]\!]$ simply returns $g^{[\text{b}/3]}$:

$$\lambda g.\{h : h \in [\![\text{he}_3 \text{ was tired}]\!](g^{[\text{b}/3]})\}$$

Assuming Bill was tired:

$$\lambda g.\{g^{[\text{b}/3]}\}$$

Suppose that our initial assignment was the empty assignment function $g_\emptyset$ (i.e. the assignment that doesn't assign any values to any numbers). Then we have the following transitions:

$$g_\emptyset \underset{\text{Bill}_3\text{ left}}{\rightsquigarrow} \{[3 \to \text{b}]\} \underset{\text{he}_3\text{ was tired}}{\rightsquigarrow} \{[3 \to \text{b}]\}$$

The discourse referent lives to fight another day! Notice that if either Bill didn't leave or wasn't tired, the evaluation gets short-circuited, and no assignments are returned.

Compare a text in a model where the text communicates something *false*. That is, assume Bill *isn't* in fact hungry (though he did leave) and interpret the following:

(2) Bill_3 left. He_3 was hungry.

In analogy with previous example, we have:

$$\lambda g.\{h : k \in [\![\text{Bill}_3 \text{ left}]\!](g) \text{ and } h \in [\![\text{he}_3 \text{ was hungry}]\!](k)\}$$

As before, given the assumption that Bill left, the left conjunct will output a single modified assignment, which means the above is equivalent to the following:

$$\lambda g.\{h : h \in [\![\text{he}_3 \text{ was hungry}]\!](g^{[\text{b}/3]})\}$$

However, given that Bill wasn't hungry, there are no assignments h output by the second conjunct. This means the above must be equivalent (in this model) to:

$$\lambda g.\emptyset$$

Supposing again that our initial assignment was the empty assignment function $g_\emptyset$, we have the following transitions:

$$g_\emptyset \underset{\text{Bill}_3\text{ left}}{\rightsquigarrow} \{[3 \to \text{b}]\} \underset{\text{he}_3\text{ was hungry}}{\rightsquigarrow} \emptyset$$

1.2 Truth

As the foregoing suggests, we can harvest truth conditions from this setup by defining truth in terms of *existential closure over outputs* (a fancy way of saying we require there to be output assignments):

$$S \text{ is true at an assignment } g \text{ iff there is an } h \text{ such that } h \in [\![S]\!](g)$$

The above notion gives us a notion of truth *simpliciter*:

$$S \text{ is true iff for every assignment } g \text{ there is an } h \text{ such that } h \in [\![S]\!](g)$$

Notice that the latter requires there to be no free variables in S (why?).

2 Indefinites

2.1 Semantics

Remember the issue with indefinites: *which* discourse referent gets introduced by e.g. *a linguist*? As it happens, the relational perspective actually offers a ready solution problem! We *don't need to make a choice*!

That is, indefinites cause a *multitude* of updated assignment functions to be output.

$$[\![\text{a linguist}_n]\!] = \lambda P.\lambda g.\bigcup\{P(x)(g^{[x/n]}) : ling'(x)\}$$

This is a new bit of notation. It's an **infinitary union**. Don't be scared. All this formula says is to do what you did for Bill, but once for each linguist, and then collect the results. For example, if the linguists are Bob and Polly, then:

$$[\![\text{a linguist}_n]\!] = \lambda P.\lambda g.P(\text{b})(g^{[\text{b}/n]}) \cup P(\text{p})(g^{[\text{p}/n]})$$

Nothing about our basic setup needs to change to accommodate this possibility. We've already assumed that sentences output sets of assignments, so everything will type out. The only difference: indefinites allow there to be potentially multiple outputs!

2.2 Example

Now let's look at an example parallel to the previous section:

(3) A linguist$_3$ left. She$_3$ was tired.

Meaning

$$[\![\text{a linguist}_3\text{ left}]\!] = \lambda g.\bigcup\left\{\{g^{[x/3]}\}\text{ if }left'(x)\text{, else }\emptyset \mid ling'(x)\right\}$$

Assume the (relevant) linguists are Anna, Irene, Polly, and Veneeta. Assume further that Irene and Polly left, but Anna and Veneeta didn't. Assume that Irene is tired, but Polly isn't.

$$\begin{aligned}[\![\text{a linguist}_3\text{ left}]\!] &= \lambda g.\emptyset \cup \{g^{[\text{i}/3]}\} \cup \{g^{[\text{p}/3]}\} \cup \emptyset \\ &= \lambda g.\{g^{[\text{i}/3]}, g^{[\text{p}/3]}\}\end{aligned}$$

Just a **nondeterminstic** variant of the previous calculation.

$$\lambda g.\{h : k \in [\![\text{a linguist}_3\text{ left}]\!](g)\text{ and }h \in [\![\text{she}_3\text{ was tired}]\!](k)\}$$

Given that Irene and Polly left:

$$\lambda g.\{h : k \in \{g^{[\text{i}/3]}, g^{[\text{p}/3]}\}\text{ and }h \in [\![\text{she}_3\text{ was tired}]\!](k)\}$$

Given that only Irene is tired:

$$\lambda g.\{g^{[\text{i}/3]}\}$$

Supposing again that our initial assignment was the empty assignment function $g_\emptyset$, we have the following transitions:

$$g_\emptyset \underset{\text{a linguist}_3\text{ left}}{\rightsquigarrow} \{[3 \to \text{i}], [3 \to \text{p}]\} \underset{\text{she}_3\text{ was tired}}{\rightsquigarrow} \{[3 \to \text{i}]\}$$

Notice how the definition of truth conspires to give us the right truth conditions, even as indefinites have received a semantics closer to that of proper names. Here, there will be an assignment output by the text iff there is a linguist who left and who was tired (here, Irene).

3 Dynamically closed things

Recall data such as (4). Assume (for the sake of simplicity) that the structure of the first sentence there is (5).

(4) I don't own a car$_i$. *It$_i$'s a Hyundai.

(5) [not [$_\Sigma$ I own a car]]

As we've seen, Σ introduces a discourse referent. This suggests that the role of negation is to, in Karttunen terms, *delete* any discourse referents introduced by its complement Σ. Here's one way to do this:

$$g[\text{not S}]h\text{ iff }g = h\text{ and }g[\text{S}] = \emptyset$$

Less iconically:

$$[\![\text{not}]\!] = \lambda p.\lambda g.\begin{cases}\{g\}\text{ if }p(g) = \emptyset \\ \text{else }\emptyset\end{cases}$$

In other words, negation both requires that its complement yields a failure (that is, has no ouputs) and that the assignment function ultimately returned is simply the unchanged input assignment. There is no chance for any drefs that may be generated in the complement to make it out alive.

Exercise: what are the implications of this sort of theory of negation for cases like (6). How about cases like (7) and (8)?

(6) I don't like Bill$_i$. He$_i$'s boring.

(7) I don't like the man who criticized Barack$_i$. He$_i$'s a good president.

(8) It isn't the case that I don't own a radio$_i$. It$_i$'s a Panasonic.

A variety of other meanings are taken to be dynamically closed:

(9) If someone$_i$ knocked, she$_i$ left. *I saw her$_i$.

(10) Harvey courts a different woman$_i$ at every convention. *She$_i$'s a climatologist.

A taste of how this goes:

$$[\![\text{if}]\!] = \lambda p.\lambda q.\,\textbf{not}(\textbf{and}(\textbf{not}(q))(p))$$
$$[\![\text{every linguist}_n]\!] = \lambda P.\,\textbf{not}(\textbf{a.ling}_n(\lambda x.\textbf{not}(P(x))))$$

4 Extensions and challenges

4.1 Abstraction

We haven't seen how we to give a semantics for abstraction along the lines of **PA**. In fact, the relevant notion is straightforward to define and basically parallel to static **PA**:

$$[\![n\ \alpha]\!] = \lambda x.\lambda g.[\![\alpha]\!](g^{[x/n]})$$

In fact, we can treat abstraction *categorematically*(!):

$$[\![n_{\langle\pi,\langle e,\pi\rangle\rangle}]\!] = \lambda\mathcal{R}.\lambda x.\lambda g.\mathcal{R}(g^{[x/n]})$$

And using this entry, we no longer need to suppose that things like *Bill* or *a linguist* introduce discourse referents themselves. Remember how in the last class I talked about abstraction in the static semantics as being a way of introducing a discourse referent? Here, we're making good on this. An example derivation is in Figure 1.

More importantly (perhaps), being able to facilitate abstraction means we can use our trusty old LF-based system for quantifier scope.

Notice that we'll need a way to make objects interpretable in situ. The types of DPs like *a linguist* and *every linguist* are (of necessity) type $\langle\langle e,\pi\rangle,\pi\rangle$, and QRing won't help because traces are also (and again of necessity) type $\langle\langle e,\pi\rangle,\pi\rangle$.

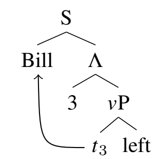
Might as well just allow in $\textsc{bla}_\emptyset$, which works the same as it did in the static semantics:

$$[\![\textsc{bla}_\emptyset]\!] = \lambda R.\lambda Q.\lambda y.Q(\lambda x.R(x)(y))$$

4.2 Destructive update

Consider the following:

(11) John$_1$ entered. Then Bill$_1$ entered. He$_?$ greeted him$_?$.



$[\![\text{S}]\!] = [\![\Lambda]\!]([\![\text{Bill}]\!])$	**FA**
$= [\![3]\!]([\![v\text{P}]\!])(\text{b})$	**FA**, Lexicon
$= (\lambda\mathcal{R}.\lambda x.\lambda g.\mathcal{R}(g^{[x/3]}))([\![v\text{P}]\!])(\text{b})$	Lexicon
$= \lambda g.[\![v\text{P}]\!](g^{[\text{b}/3]})$	$\beta\times 2$
$= \lambda g.[\![t_3]\!]([\![\text{left}]\!])(g^{[\text{b}/3]})$	**FA**
$= \lambda g.(\lambda P.\lambda g.P(g(3))(g))(\textbf{left})(g^{[\text{b}/3]})$	**FA**
$= \lambda g.\textbf{left}(g^{[\text{b}/3]}(3))(g^{[\text{b}/3]})$	$\beta\times 2$
$= \lambda g.\textbf{left}(\text{b})(g^{[\text{b}/3]})$	$\equiv$
$= \lambda g.\{g^{[\text{b}/3]}\}$ if $left'(\text{b})$, else $\emptyset$	Definition, $\beta\times 2$

Figure 1: Abstraction-induced drefs

Suppose both John and Bill entered, and that the beginning assignment is the empty assignment $g_\emptyset$. Then the update chain will look as follows:

$$g_\emptyset \underset{\text{John}_1\text{ entered}}{\rightsquigarrow} \{[1\rightarrow\text{j}]\} \underset{\text{Bill}_1\text{ entered}}{\rightsquigarrow} \{[1\rightarrow\text{b}]\}$$

So because we weren't careful about choosing our indices, we've managed to overwrite the first one. That is, we've lost some information as we've gone along.

But why the heck should this trouble us? Remember the same sort of thing happened when we had two abstraction indices competing. In that case, the lower one took precedence over the higher one (also how the lambda calculus works!).

Remarkably, a lot of energy has been devoted to avoiding this "issue". Why? Perhaps dynamic semantics is supposed to be modeling information growth. In that sense, we would, perhaps, expect not to have updates that destroy information.

4.3 Final issues

Consider the implications of the above analysis for donkey anaphora, inverse linking, vis à vis ellipsis. Imagine accounting for *ellipsis* itself in terms of dynamic anaphora!

Consider the status of incorporating linear asymmetries into the semantics, in light of the fact that crossover plausibly has a linearity component.

There's a nearby version of this semantics that gets by without co-indexation between antecedent and anaphor. If we have time, we'll see how it works.