Dynamic semantics for indefinites

March 11, 2015

1 Today

- We'll learn a bit about the treatment of "exceptional" varieties of anaphora within dynamic semantics.
- Of particular interest to us: the dynamic analysis of indefinites (very interesting), and the way that it manages composition (extremely boring).
- We will see that dynamic semantics for anaphora has many points of contact with alternative semantics. For instance, both say that the meanings of indefinites correspond in some sense to sets — that is, both task indefinites with *introducing alternatives*. Moreover, both approaches rely on *closure operators* to tame the alternatives invoked by indefinites (and to extract truth conditions).
- At the same time, the relation between dynamic and alternative semantics will remain somewhat unclear. The former but not the latter handles exceptional binding. The latter but not the former handles exceptional scope. And the sorts of alternatives they deal with are fundamentally different in kind.
- Given that, is the appearance of unity a mirage, or are there prospects for an integrated theory, with the best of both worlds? And how does all of this relate to the questions we've been considering in previous weeks (e.g. exceptional scope, selectivity, etc.)?
- The discussion will, of necessity, be somewhat technical. So **please** alert me to points of unclarity or undue speed.

2 Anapahora to quantifiers?

- Coreference without binding:

(1) a. Barack$_i$ came in. He$_i$ sat down.
  b. Everyone who admires Clinton$_i$ voted for her$_i$.

- Why *not* binding? It would seem to rule in the following:

(2) a. *Nobody$_i$ came in. He$_i$ sat down.
  b. *Everyone who admires nobody$_i$ voted for her$_i$.

- But of course no binding is no trouble. We can just interpret the unbound pronouns in (1) via the contextually given assignment function. However, other cases seem a bit tougher, in that they seem to involve exceptional anaphora to a *quantifier*:

(3) a. A senator$_i$ admires Kennedy. He$_i$'s very junior.
  b. Exactly one senator$_i$ admires Kennedy. He$_i$'s very junior.
  c. Every farmer who owns a donkey$_i$ beats it$_i$.

- Again, this can't be binding in the standard Heim & Kratzer 1998 sense. Suppose it was. This would imply that example (3-b) was generated by an LF like the one in Figure 1. But this LF is disastrously made true if there's many senators who admire Kennedy, but just one who's very junior. (To say nothing of how you'd handle examples like (3-c).)

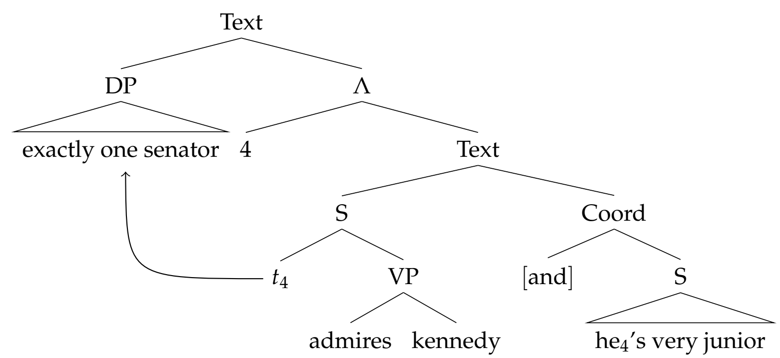

Figure 1: Disastrous LF for example (3-b).

- A natural thought is that the pronouns in these cases are covert descriptions, e.g. that the pronoun in examples like (3-a) and (3-b) is proxy for something like *the senator who admires Kennedy*. On this analysis, the appearance of "binding" is only apparent.

- But it is a challenge to work this out. Notice for instance that this would seem to imply that the pronoun's referent is the unique individual satisfying the description. But binding in such examples remains felicitous when there are multiple individuals satisfying the relevant description.

(4) a. Al has a dog$_i$. He feeds it$_i$ tasty morsels. Al has another dog$_j$, but he only feeds it$_j$ scraps. (Abbott 2002)
b. A man$_i$ walked into the room with another man. Everyone turned to look at him$_i$.
c. Everybody who buys a sage plant$_i$ here buys eight others along with it$_i$. (Heim 1982)

- We'll explore an alternative line: that certain expressions can introduce **discourse referents** (Karttunen 1976).

3 A dynamic formal language

- Basic idea: a discourse *state* encodes information about the discourse referents available.
- The system we'll sketch here (essentially Dekker 1994's PLA) has a number of interesting properties, but I've mostly picked it because it's (in my view) maximally simple. Some additional semantics doing essentially the same thing: Groenendijk & Stokhof 1991; Heim 1982; Kamp 1981; Barwise 1987; Rooth 1987.
- PLA formulas are in the business of *mapping states to states*, possibly adding discourse information in the process. (This is the sense in which this semantics is *dynamic*.)
- A state, in turn, is modeled as a set of *sequences of discourse referents*. For example, here is a state with 5 discourse referents; it exhibits **certainty** about the middle and last discourse referent (resp. c and e), and **uncertainty** about the remaining three discourse referents.

$$\left\{ \begin{array}{l} \mathsf{a\,b\,\boxed{c}\,d\,\boxed{e}}, \\ \mathsf{b\,b\,\boxed{c}\,e\,\boxed{e}}, \\ \mathsf{a\,a\,\boxed{c}\,b\,\boxed{e}} \end{array} \right\}$$

- The sequences that comprise a state can be extended...

$$\omega ... cba \cdot x = \omega ... cbax$$

- ...and mined for drefs (e.g. if $e = \mathsf{abcde}$, $e_0 = \mathsf{e}$, and $e_3 = \mathsf{b}$):

$$e_n = \text{the } n^{th}\text{-to-last member of } e$$

- This suggests a notion of information growth, very much like the sort of information growth you find in Stalnakerian approaches to the common ground. Information growth can happen in one of two ways: (1) you eliminate possibilities, i.e. reduce uncertainty, or (2) you learn about some new discourse referents. More formally:

$$e \leqslant e' \text{ iff } \exists e''.\, e' = e \cdot e'' \qquad s \leqslant s' \text{ iff } \forall e' \in s'.\, \exists s \in s.\, e \leqslant e'$$

- Information growth comes about via updates with formulas. One way to think of this is that formulas denote functions from states into new states with at least at much information. If p is a formula meaning and s is an incoming state, we will notate the update of s by p, i.e. $p(s)$, iconically (if somewhat confusingly) as follows:

$$s[p]$$

- Syntax: Predicate Logic (with equality), plus a new category of terms — i.e. pronouns $A = \{\mathsf{p}_i : i \in \mathbb{N}\}$. Thus, in addition to the usual well-formed formulae (e.g. $\exists x \phi$, $(\phi \wedge \psi)$, $\neg\phi$, and so on), the following sorts of expressions are well-formed:

$$\mathsf{p}_0 = 5 \qquad (\exists x.\, x = \mathsf{p}_3) \wedge \mathsf{odd}(\mathsf{p}_0)$$

- Semantics:
  - $s[\![Rt_1 ... t_n]\!]^g = \{e \in s : \langle [t_1]_{e,g}, ..., [t_n]_{e,g} \rangle \in [\![R]\!]\}$
  - $s[\![t_1 = t_2]\!]^g = \{e \in s : [t_1]_{e,g} = [t_2]_{e,g}\}$
  - $s[\![\neg\phi]\!]^g = \{e \in s : \{e\}[\![\phi]\!]^g = \varnothing\}$
  - $s[\![\exists v \phi]\!]^g = \{e' \cdot d : d \in \mathcal{D} \wedge e' \in s[\![\phi]\!]^{g[v \to d]}\}$
  - $s[\![\phi \wedge \psi]\!]^g = s[\![\phi]\!]^g[\![\psi]\!]^g$

- The key bit here is the semantics for existential quantification. Notice that it is expressed in terms of *state extension*!
- Notice also that conjunction is expressed in terms of successive updates. If you actually work out what this means, given the update notation, it corresponds to *function composition*, i.e. $[\![\psi]\!]^g([\![\phi]\!]^g(s))$. This a little mind-warping.
- Semantics for terms. Constants are constant, variables are evaluated relative to the assignent function, and pronouns look to a sequence for their meaning (remember that the sequence is the sort of thing that holds discourse referents).
  - $[c]_{e,g} = [\![c]\!]$, for all constants c
  - $[v]_{e,g} = g(v)$, for all variables v
  - $[\mathrm{p}_n]_{e,g} = e_n$, for all pronouns p_n
- Some examples [see `ghci` demo].
- How about truth conditions? It is straightforward to define a derivative notion of truth, corresponds to a non-failed update (consider how we might define a truth-condition-extracting operator in alternative semantics).

$$p \text{ is true at } s \text{ iff } s[p] \neq \emptyset$$

4 Going Montagovian

- Suppose, with PLA, that sentences denote *update functions on states*, type $\sigma \to \sigma$. Call this new type $\mathbf{T} ::= \sigma \to \sigma$.
- To build a compositional dynamic semantics, systematically replace t in expressions' types with $\mathbf{T}$. Thus, the type of a verb phrase becomes $e \to \mathbf{T}$, the type of a transitive verb becomes $e \to e \to \mathbf{T}$, the type of a quantifier becomes $(e \to \mathbf{T}) \to \mathbf{T}$, and so on..
- The meanings associated with predicates and relations simply test that their arguments are in the extension of the normal predicate/relation. If that is so, they return the initial state. Otherwise, they produce a failure, i.e. $\emptyset$:

$$\mathbf{left} = \lambda x.\, \lambda s. \begin{cases} s \text{ if } x \text{ left} \\ \text{otherwise } \emptyset \end{cases} \tag{1}$$

$$\mathbf{met} = \lambda x.\, \lambda y.\, \lambda s. \begin{cases} s \text{ if } y \text{ likes } x \\ \text{otherwise } \emptyset \end{cases} \tag{2}$$

Notice that **left** is type $e \to \mathbf{T}$ and **met** is type $e \to e \to \mathbf{T}$.

- Indefinites and names are just taken to introduce discourse referents. Both of these are type $(e \to \mathbf{T}) \to \mathbf{T}$.

$$\mathbf{Polly} = \lambda P.\, \lambda s.\, P(\mathsf{p})(s \cdot \mathsf{p}) \tag{3}$$

$$\mathbf{a.linguist} = \lambda P.\, \lambda s. \bigcup_{x \in \mathsf{ling}} P(x)(s \cdot x) \tag{4}$$

- ...Where $s \cdot a$ just means adding a to the sequences already in s:

$$s \cdot a = \{e \cdot a : e \in s\}$$

- The meaning for *a linguist* can be equivalently rewritten:

$$\lambda P.\, \lambda s.\, \{e' : e \in s \wedge \\ x \in \mathsf{ling} \wedge \\ e' \in P(x)(e \cdot x)\}$$

- Similarly, pronouns just retrieve discourse referents:

$$\mathbf{pro}_n = \lambda P.\, \lambda s. \bigcup\{P(e_n)(e) : e \in s\} \tag{5}$$

- The compositional apparatus is quite spare. For example, since **left** is type $e \to \mathbf{T}$, and **Polly** is type $(e \to \mathbf{T}) \to \mathbf{T}$, the latter can apply to the former via garden-variety functional application.
- Notice: this semantics differs very slightly from PLA, in that there are no assignment functions. It's just natural to do without them in this setting.

- Suppose for illustration that our discourse begins with an initial state init, a singleton set containing only the empty sequence:

$$\mathsf{init} = \{\langle\rangle\}$$

- Here's an example with a proper name, assuming *Polly left* is true:

$$\mathsf{init}[\![\text{Polly left}]\!] \longrightarrow \{\mathsf{p}\}$$

- Correspondingly, here is an example with an indefinite, assuming that the linguists who left are Al and Betty:

$$\mathsf{init}[\![\text{a linguist left}]\!] \longrightarrow \{\mathsf{a}, \mathsf{b}\}$$

- As in PLA, conjunction is just sequencing, i.e. successive update:

$$\textbf{and} = \lambda R.\, \lambda L.\, \lambda S.\, S[L][R] \tag{6}$$

  And, as before, the right-hand side here is equivalent, somewhat confusingly at first, to $R(L(S))$.

- Example: sequencing *Polly left* and *a linguist left*:

$$\mathsf{init}[\![\text{Polly left}]\!][\![\text{a linguist left}]\!] \longrightarrow= \{\mathsf{pa}, \mathsf{pb}\}$$

- Adding in a pronoun. Assume that linguist a is tired, but linguist b is not. Then we have the following transition:

$$\mathsf{init}[\![\text{a linguist left}]\!][\![\text{she}_0 \text{ was tired}]\!] = \{\mathsf{a}\}$$

5 Dynamically closed meanings

- Certain meanings (e.g. negation and universals) are less congenial to discourse reference. Not only do they kill off any discourse referents generated in their scope — examples (5-a) and (5-b) — they also fail to generate any discourse referents of their own — example (5-c).

(5) a. I don't own a car$_i$. *It$_i$'s a Mazda. $(\neg > \exists)$
  b. Everyone met a phonologist$_i$. *He$_i$ was tall. $(\forall > \exists)$
  c. John met every linguist$_i$. *She$_i$ was tall.

- We begin by defining a meaning for negation. It knocks out any points in the incoming state that lead to a successful update.

$$\textbf{not} = \lambda p.\, \lambda s.\, \{e \in s : \{e\}[p] = \varnothing\} \tag{7}$$

  Equivalently:

$$\lambda p.\, \lambda s.\, \{e \in s : \neg\exists e'.\, e \leqslant e' \wedge e' \in s[p]\}$$

- Negation is a kind of closure operator. It quantifies over and then discards any alternatives and discourse referents generated in its scope. Along these lines, we might define an *existential* closure operator as follows:

$$\begin{aligned} \mathcal{E} &= \lambda p.\, \textbf{not}(\textbf{not}(p)) \\ &= \lambda p.\, \lambda s.\, \{e \in s : \{e\}[p] \neq \varnothing\} \end{aligned} \tag{8}$$

  The function of $\mathcal{E}$ is to eliminate any discourse referents that might be generated in p, while retaining p's truth conditions. The upshot is that p's truth-conditional content is inherited by the new update function, but its discourse referents are discarded.

- Similarly, we also have dynamically closed generalized quantifiers (to keep things as simple as possible, I'm ignoring the fact that these quantifiers introduce a discourse referent to their scope; the actual semantics replaces the right-most s with $s \cdot x$).

$$\textbf{every.linguist} = \lambda P.\, \lambda s. \begin{cases} s \text{ if } \forall x \in \mathsf{ling}.\, s[P(x)] \neq \varnothing \\ \text{otherwise } \varnothing \end{cases} \tag{9}$$

$$\textbf{no.linguist} = \lambda P.\, \lambda s. \begin{cases} s \text{ if } \forall x \in \mathsf{ling}.\, s[P(x)] = \varnothing \\ \text{otherwise } \varnothing \end{cases} \tag{10}$$

  Notice e.g. that $s[P(x)] \neq \varnothing$ iff $\exists e.\, e \in s[P(x)]$. That is, these meanings build in a kind of existential closure. Note: each of these meanings can be defined in terms of negation (and this is how these definitions are standardly stated in dynamic semantics). For example, **every.linguist** $= \lambda P.\, \textbf{not}(\textbf{a.linguist}\ (\lambda x.\, \textbf{not}(P(x))))$.

- Explains why discourse referents generated in the scope of quantifiers and negation seem to die off. The function of **not** and **every.linguist** is to check that their arguments satisfy certain properties, and then to toss out any discourse referents generated in the process — returning the input state unchanged if the procedure succeeds.[1]

- In sum, there is a push and a pull in dynamic semantics. Certain meanings are in the business of creating discourse referents, and certain meanings are in the business of eliminating them.

6 Connecting to Alternatives

- There are some remarkable similarities between dynamic and alternative semantics.
- In both dynamic and alternative semantics, indefinites do what proper names do, only multiple times. The contribution of an indefinite to the meaning of a sentence thus consists, in both cases, in the generation of alternatives!
- In dynamic semantics, the alternatives are sequences of discourse referents. In alternative semantics, the alternatives can be any garden-variety meanings (e.g. individuals, properties, propositions, etc.).
- In both cases, closure operators discharge alternatives. In dynamic semantics, closure is built into the semantics of negation and quantification. What this means is that, whenever an alternative generator is caught within the scope of a closure operator Op, its alternatives will be invisible outside the scope of Op.
- At the same time, there are some important *disunities*.
- Unlike alternative semantics, in dynamic semantics the alternatives are not handled via a new composition operation. They're just baked into the semantics. Dynamic composition is extremely boring, and relies only on functional application. Notice, for example, that there's no abstraction issue in dynamic semantics: since we never need to go from sets of propositions to set of functions (unlike alternative semantics!), we're in the clear.
- In dynamic semantics the alternatives only exist at the propositional level. In alternative semantics, it's "alternatives all the way down." Correspondingly, in dynamic semantics, a false sentence won't have *any* alternatives. Falsity just corresponds to failure, i.e. an empty result state. By contrast, in alternative semantics, false propositions can make it into an alternative set.
- And, perhaps most basically, they're just geared to do different things. Dynamic semantics is all about exceptional binding. Alternative semantics is (in part) all about exceptional scope. But wouldn't it be nice to unify the two? And might doing so shed any light on the issues we've been considering?

[1]We likewise explain why *every linguist* and *no linguist* don't bind out of their sentences, even on the dref-generating meanings for the quantifiers. Do you see why?

References

Abbott, Barbara. 2002. Donkey Demonstratives. *Natural Language Semantics* 10(4). 285–298. doi:10.1023/A:1022141232323.

Barwise, Jon. 1987. Noun Phrases, Generalized Quantifiers, and Anaphora. In Peter Gärdenfors (ed.), *Generalized Quantifiers*, 1–29. Dordrecht: Reidel.

Dekker, Paul. 1994. Predicate Logic with Anaphora. In Mandy Harvey & Lynn Santelmann (eds.), *Proceedings of Semantics and Linguistic Theory 4*, 79–95. Ithaca, NY: Cornell University.

Groenendijk, Jeroen & Martin Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy* 14(1). 39–100. doi:10.1007/BF00628304.

Heim, Irene. 1982. *The Semantics of Definite and Indefinite Noun Phrases*: University of Massachusetts, Amherst Ph.D. thesis.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar*. Oxford: Blackwell.

Kamp, Hans. 1981. A theory of truth and semantic interpretation. In Jeroen Groenendijk, Theo Janssen & Martin Stokhof (eds.), *Formal Methods in the Study of Language*, 277–322. Mathematical Centre Amsterdam.

Karttunen, Lauri. 1976. Discourse referents. In James D. McCawley (ed.), *Syntax and Semantics, volume 7: Notes from the Linguistic Underground*, 363–385. New York: Academic Press.

Rooth, Mats. 1987. Noun Phrase Interpretation in Montague Grammar, File Change Semantics, and Situation Semantics. In Peter Gärdenfors (ed.), *Generalized Quantifiers*, 237–269. Dordrecht: Reidel.