PLA, compositional dynamics

Semantics II

April 9, 2018

Predicate Logic with Anaphora

File Change Semantics, or close enough (Heim 1982, 1983)

$$
\begin{aligned}
s[R x_1 \ldots x_n] &= \{g \in s \mid ([\![x_1]\!]^g, \ldots, [\![x_n]\!]^g) \in [\![R]\!]^g\} \\
s[\phi \wedge \psi] &= s[\phi][\psi] \\
s[\neg\phi] &= \{g \in s \mid \{g\}[\phi] = \emptyset\} \\
s[\exists^i] &= \{g \cup (i, x) \mid g \in s, x \in D\}
\end{aligned}
$$

Illustration

$$
\begin{array}{c}
s \\
\Big\downarrow \ \exists^1 \wedge \mathbf{man}\, x_1 \wedge \mathbf{witp}\, x_1 \\
\{g \cup (1, x) \mid g \in s, \mathrm{man}\, x, \mathrm{witp}\, x\} \\
\Big\downarrow \ \mathbf{whistled}\, x_1 \\
\{g \cup (1, x) \mid g \in s, \mathrm{man}\, x, \mathrm{witp}\, x, \mathrm{whistled}\, x\} \\
\Big\downarrow \ \exists^2 \wedge \mathbf{woman}\, x_2 \\
\{g \cup (1, x) \cup (2, y) \mid g \in s, \mathrm{man}\, x, \mathrm{witp}\, x, \mathrm{whistled}\, x, \mathrm{woman}\, y\}
\end{array}
$$

Dynamic properties of FCS

FCS is non-eliminative: processing a context does not always lead us to a state that is a subset of that context. In particular, dynamic existential quantifiers can *expand* the context to include new anaphoric possibilities.

As such, we should expect some non-trivially dynamic behavior in the system. And we get it. FCS conjunction is *non-commutative*. That is, in general:

$$\phi \wedge \psi \neq \psi \wedge \phi$$

So order matters. We also observe failures of *idempotence*:

$$\phi \neq \phi \wedge \phi$$

PLA syntax and semantics

The core syntax of PLA is exactly that of first-order logic:

$$\text{Atom} ::= R\,t_1 \ldots t_n \qquad \phi ::= \text{Atom} \mid \phi \wedge \phi \mid \neg\phi \mid \exists v : \phi$$

PLA semantics is conservative, but existential quantifiers *extend* the sequence:

$$\begin{aligned}
s[R\,t_1 \ldots t_n]^g &= \{e \in s \mid (\llbracket t_1 \rrbracket^{e,g}, \ldots, \llbracket t_n \rrbracket^{e,g}) \in \llbracket R \rrbracket^{e,g}\} \\
s[\phi \wedge \psi]^g &= s[\phi]^g[\psi]^g \\
s[\neg\phi]^g &= \{e \in s \mid \{e\}[\phi]^g = \emptyset\} \\
s[\exists x : \phi]^g &= \{\underbrace{e \cdot d}_{\text{extending } e \text{ with the } d\text{'s that make } \phi \text{ true}} \mid d \in D, e \in s[\phi]^{g[x \to d]}\}
\end{aligned}$$

Terms

The *terms* of PLA are slightly more articulated than standard FOL:

$$\texttt{Term} ::= \underbrace{x \mid y \mid \ldots}_{\text{Variables}} \mid \underbrace{\mathsf{p}_0 \mid \mathsf{p}_1 \mid \ldots}_{\text{Pronouns}}$$

Variables and pronouns are evaluated in different ways:

$$[\![x]\!]^{e,g} = g_x \qquad [\![\mathsf{p}_n]\!]^{e,g} = \text{the } n\text{-th member of } e$$

Variables are standard: their value is determined by the assignment. Pronoun semantics is determined by the *sequence* constructed in interpretation.

Example calculation

$$\begin{aligned}
s[\exists x : \mathbf{man}\, x]^g &= \{e \cdot d \mid d \in D, e \in s[\mathbf{man}\, x]^{g[x \to d]}\} \\
&= \{e \cdot d \mid d \in D, e \in \{e' \in s \mid [\![x]\!]^{e,g[x \to d]} \in [\![\mathbf{man}]\!]^{e,g[x \to d]}\}\} \\
&= \{e \cdot d \mid d \in D, e \in \{e' \in s \mid \mathrm{man}\, d\}\} \\
&= \{e \cdot d \mid d \in D, e \in s, \mathrm{man}\, d\}
\end{aligned}$$

Suppose our domain has 3 individuals, $D = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$, and that a and c are linguists, and only a is French. Then we may observe the following succession of updates:

$$
\begin{tikzcd}
\{\emptyset\} \arrow[d, dotted, "\exists x : \textbf{ling}\, x"] \\
\{\mathrm{a}, \mathrm{c}\} \arrow[d, dotted, "\textbf{french}\, \mathrm{p}_0"] \\
\{\mathrm{a}\} \arrow[d, dotted, "\exists x : \textbf{ling}\, x"] \\
\{\mathrm{aa}, \mathrm{ac}\} \arrow[d, dotted, "\neg\textbf{french}\, \mathrm{p}_1"] \\
\{\mathrm{ac}\} \arrow[d, dotted, "\exists x : \neg\textbf{ling}\, x"] \\
\{\mathrm{acb}\} \\
\vdots
\end{tikzcd}
$$

Info states in PLA

PLA info-states carry information about which individuals have been made salient in a discourse. More to the point, they carry *those individuals*, in the order they were introduced. Uncertainty about a referent corresponds to variation across a column:

$$\left\{\begin{array}{l} \mathrm{ab{\color{orange}c}d{\color{blue}e}}, \\ \mathrm{bb{\color{orange}c}e{\color{blue}e}}, \\ \mathrm{aa{\color{orange}c}b{\color{blue}e}} \end{array}\right\}$$

Information growth happens in two ways: (1) you eliminate possibilities, i.e. reduce uncertainty, or (2) you learn about some new discourse referents.

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

Variable-free dynamics

PLA is motivated mostly by a desire for a dynamic semantics that *conservatively extends* static first-order logic with apparatus for pronouns.

But it gives us the resources to dispense with assignments and variables entirely:

$$\mathtt{Atom} ::= R\,t_1 \ldots t_n \qquad \phi ::= \mathtt{Atom} \mid \phi \wedge \phi \mid \neg\phi \mid \exists$$

The semantics is purely about generating and propagating verifying sequences:

$$\begin{aligned}
s[R\,t_1 \ldots t_n] &= \{e \in s \mid ([\![t_1]\!]^e, \ldots, [\![t_n]\!]^e) \in [\![R]\!]^e\} \\
s[\phi \wedge \psi] &= s[\phi][\psi] \\
s[\neg\phi] &= \{e \in s \mid \{e\}[\phi] = \emptyset\} \\
s[\exists] &= \{e \cdot d \mid e \in s, d \in D\}
\end{aligned}$$

Illustration

$$\begin{tikzcd}
\{\emptyset\} \arrow[d, dotted, "\exists"] \\
\{\mathbf{a}, \mathbf{b}, \mathbf{c}\} \arrow[d, dotted, "\textbf{french}\ \mathsf{p}_0"] \\
\{\mathbf{a}, \mathbf{c}\} \arrow[d, dotted, "\exists"] \\
\{\mathbf{aa}, \mathbf{ab}, \mathbf{ac}, \mathbf{ca}, \mathbf{cb}, \mathbf{cc}\}
\end{tikzcd}$$

Compositional dynamics

From updates to relations

Our dynamic meanings for our formal language were *update functions on contexts*, with contexts rendered as sets of "anaphoric possibilities". E.g., for FCS:

$$\mathrm{T} ::= \underbrace{\{\mathrm{g}\} \to \{\mathrm{g}\}}_{\text{dynamic propositions are update functions}}$$

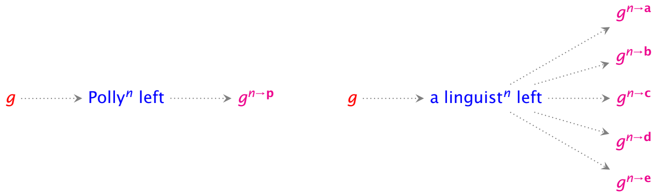
Given that these update functions are always distributive (they process the input set point-wise), we can just as well think of them as *relations*:

$$\mathrm{T} ::= \underbrace{\mathrm{g} \to \{\mathrm{g}\}}_{\text{dynamic propositions are update relations}}$$

Comparing the two perspectives, we see there's no real difference:

$$s[\exists^i] = \{g \cup (i, x) \mid g \in s, x \in D\} \qquad g[\exists^i] = \{g \cup (i, x) \mid x \in D\}$$

The basic idea[1]



- Dref introduction is assignment modification.
- Indefinites introduce drefs *non-deterministically*.
- New drefs may (not) pan out downstream (cf. Stalnaker 1978).

[1] Heim (1982), Barwise (1987), Rooth (1987), Groenendijk & Stokhof (1991), Muskens (1996), etc.

Relation composition

When sentence meanings are update functions, successive update (i.e., conjunction) can be modeled via function composition:

$$\begin{aligned} [\phi \wedge \psi] &= \lambda s.\, [\psi]\,([\phi]\, s) \\ &= [\psi] \circ [\phi] \\ &= [\phi]\, ;\, [\psi] \end{aligned}$$

If sentence meanings are update *relations*, this needs to be adjusted: $[\psi]\,([\phi]\, s)$ isn't well-typed, since $[\phi]\, s$ is a set, but $[\psi]$ expects a single thing as input.

Can you figure out how to compose relations instead of functions?

Relation composition

When sentence meanings are update functions, successive update (i.e., conjunction) can be modeled via function composition:

$$\begin{aligned} [\phi \wedge \psi] &= \lambda s.\, [\psi]\,([\phi]\, s) \\ &= [\psi] \circ [\phi] \\ &= [\phi]\, ;\, [\psi] \end{aligned}$$

If sentence meanings are update *relations*, this needs to be adjusted: $[\psi]\,([\phi]\, s)$ isn't well-typed, since $[\phi]\, s$ is a set, but $[\psi]$ expects a single thing as input.

Can you figure out how to compose relations instead of functions?

$$R \circ L = \lambda g. \bigcup_{h \in L g} R\, h$$

Dynamic sentence meanings

Equivalently, we can think of relations as sets of ordered pairs:

$$a \to \{a\} \simeq a \to a \to \mathtt{t} \simeq \{a \times a\}$$

Our standard semantic values (ignoring intensionality) can be conceived of as *sets of verifying assignment functions*, type $\{\mathtt{g}\}$ (recall $[\![\cdot]\!]_{¢}$ from intensional semantics):

$$[\![\text{she}_3 \text{ whistled}]\!]_{¢} = \{g \mid g_3 \in \text{whistled}\}$$

Going dynamic means thinking of sentence meanings as relations on assignment functions, i.e., as *sets of pairs of assignments*.

Static propositions: $\{\underbrace{\mathtt{g}}_{\text{input}}\}$ Dynamic propositions: $\{\underbrace{\mathtt{g}}_{\text{input}} \times \underbrace{\mathtt{g}}_{\text{output}}\}$

So sentences aren't just sensitive to the assignment — they can also *change* it.

Dynamic semantics in two steps

Sentences denote updates, i.e., relations on assignments:

$$\mathsf{T} ::= \mathsf{g} \to \{\mathsf{g}\} \qquad\qquad [\![\text{left}]\!] := \underbrace{\lambda x.\lambda g.\,\{g \mid x \in \text{left}\}}_{\mathsf{e}\to\mathsf{T}}$$

Precisely the same thing goes for a transitive verb:

$$[\![\text{likes}]\!] := \underbrace{\lambda x.\lambda y.\lambda g.\,\{g \mid \text{likes}\, x\, y\}}_{\mathsf{e}\to\mathsf{e}\to\mathsf{T}}$$

Practice: what meaning is assigned by this theory to *Polly likes Chris*?

Dynamic semantics in two steps

Sentences denote updates, i.e., relations on assignments:

$$\mathrm{T} ::= \mathrm{g} \to \{\mathrm{g}\} \qquad\qquad [\![\text{left}]\!] := \underbrace{\lambda x.\lambda g.\,\{g \mid x \in \text{left}\}}_{\mathrm{e}\to\mathrm{T}}$$

Precisely the same thing goes for a transitive verb:

$$[\![\text{likes}]\!] := \underbrace{\lambda x.\lambda y.\lambda g.\,\{g \mid \text{likes}\, x\, y\}}_{\mathrm{e}\to\mathrm{e}\to\mathrm{T}}$$

Practice: what meaning is assigned by this theory to *Polly likes Chris*?

$$\begin{aligned}
[\![\text{Polly [likes Chris]}]\!] &= [\![\text{likes Chris}]\!]\,[\![\text{Polly}]\!] && \text{FA}\\
&= [\![\text{likes}]\!]\,[\![\text{Chris}]\!]\,[\![\text{Polly}]\!] && \text{FA}\\
&= (\lambda x.\lambda y.\lambda g.\,\{g \mid \text{likes}\, x\, y\})\,\mathrm{c}\,\mathrm{p} && \text{Lex} \times 3\\
&= \lambda g.\,\{g \mid \text{likes}\,\mathrm{c}\,\mathrm{p}\} && \beta \times 2
\end{aligned}$$

If Polly likes Chris, the input g is returned. If she *doesn't*, nothing is returned.

Static vs. dynamic perspectives

In a static semantics, *Polly likes Chris* denotes a set of assignments. It's either the *full* set of all assignments or the empty set of none, depending on the facts.

$$[\![\text{Polly likes Chris}]\!]_{¢} = \{g \mid \mathsf{likes}\,\mathsf{c}\,\mathsf{p}\}$$

In a dynamic semantics, *Polly likes Chris* denotes a relation on assignments. It's either the *identity* relation, or the empty relation, depending on the facts.

$$[\![\text{Polly likes Chris}]\!] = \lambda g.\,\{g \mid \mathsf{likes}\,\mathsf{c}\,\mathsf{p}\} \simeq \{(g, g) \mid \mathsf{likes}\,\mathsf{c}\,\mathsf{p}\}$$

Binding

So that's how simple sentences work. If we want a rule like Predicate Modification, we'll have to state one that works with dynamic propositions, but that is straightforward to do, and we pass over it today.

Of more immediate interest is the dynamic correlate of *Predicate Abstraction*:

Static PA: $[\![n\ \alpha]\!]^g = \lambda x.\,[\![\alpha]\!]^{g[n \to x]}$

It turns out that we can make fairly direct use of this rule in a dynamic system:

Dynamic PA:

Binding

So that's how simple sentences work. If we want a rule like Predicate Modification, we'll have to state one that works with dynamic propositions, but that is straightforward to do, and we pass over it today.

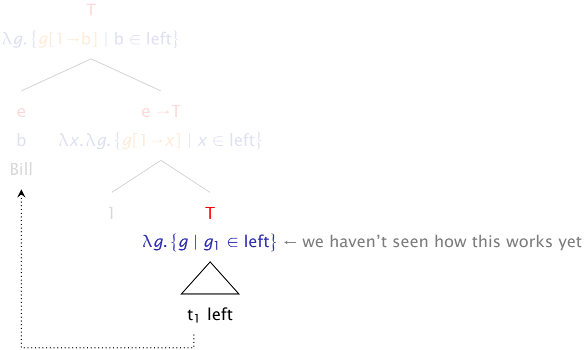
Of more immediate interest is the dynamic correlate of *Predicate Abstraction*:

$$\text{Static PA: } [\![n\ \alpha]\!]^{g} = \lambda x.\, [\![\alpha]\!]^{g[n \to x]}$$

It turns out that we can make fairly direct use of this rule in a dynamic system:

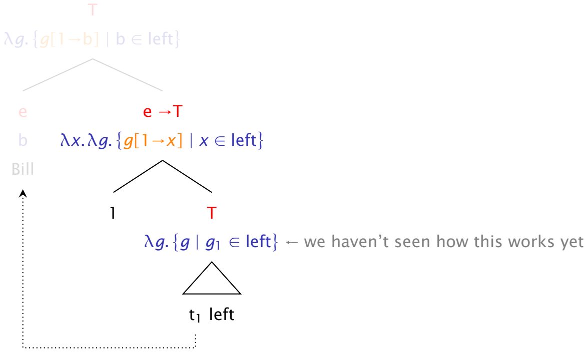
$$\text{Dynamic PA: } [\![n\ \alpha]\!] = \lambda x.\lambda g.\, [\![\alpha]\!]\, g[n \to x]$$

Example: *Bill t left*



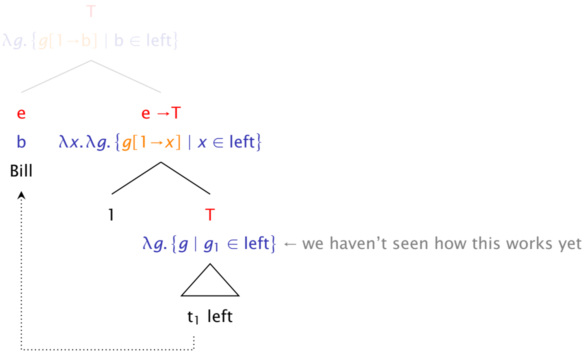
It isn't exaggerating to say that this is *the key feature of dynamic semantics*: when the assignment function shifts, perhaps even via a run-of-the-mill Predicate Abstraction, the shifted assignments are *remembered above the point of the shift*.

Example: *Bill t left*



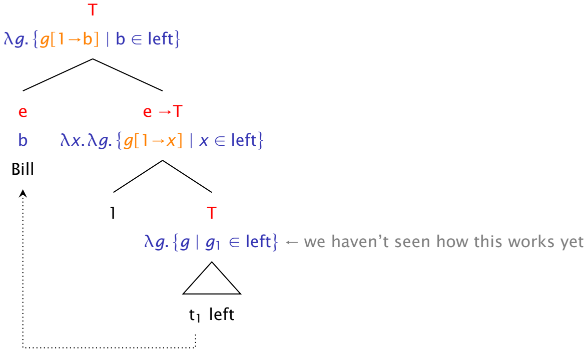
It isn't exaggerating to say that this is *the key feature of dynamic semantics*: when the assignment function shifts, perhaps even via a run-of-the-mill Predicate Abstraction, the shifted assignments are *remembered above the point of the shift*.

Example: *Bill t left*



It isn't exaggerating to say that this is *the key feature of dynamic semantics*: when the assignment function shifts, perhaps even via a run-of-the-mill Predicate Abstraction, the shifted assignments are *remembered above the point of the shift*.

Example: *Bill t left*



It isn't exaggerating to say that this is *the key feature of dynamic semantics*: when the assignment function shifts, perhaps even via a run-of-the-mill Predicate Abstraction, the shifted assignments are *remembered above the point of the shift*.

Traces and pronouns

What kind of dynamic-semantics should we give to traces and pronouns? Solve for ?:

$$\underbrace{?}_{[\![\mathrm{t}_1]\!]} + \underbrace{\lambda x.\lambda g.\{g \mid x \in \mathrm{left}\}}_{[\![\mathrm{left}]\!]} = \lambda g.\{g \mid g_1 \in \mathrm{left}\}$$

Actually, this fully determines $[\![\mathrm{t}_1]\!]$. What should it be?

Traces and pronouns

What kind of dynamic-semantics should we give to traces and pronouns? Solve for ?:

$$\underbrace{?}_{[\![t_1]\!]} + \underbrace{\lambda x.\lambda g.\{g \mid x \in \text{left}\}}_{[\![\text{left}]\!]} = \lambda g.\{g \mid g_1 \in \text{left}\}$$

Actually, this fully determines $[\![t_1]\!]$. What should it be?

$$[\![t_1]\!] = \underbrace{\lambda f.\lambda g.f\,g_1\,g}_{(e\to T)\to T}$$

This in turn implies something a little strange about the semantics of transitive verbs. They cannot be type $e \to e \to T$. Why not?

Traces and pronouns

What kind of dynamic-semantics should we give to traces and pronouns? Solve for ?:

$$\underbrace{?}_{[\![t_1]\!]} + \underbrace{\lambda x.\lambda g.\{g \mid x \in \mathsf{left}\}}_{[\![\mathsf{left}]\!]} = \lambda g.\{g \mid g_1 \in \mathsf{left}\}$$

Actually, this fully determines $[\![t_1]\!]$. What should it be?

$$[\![t_1]\!] = \underbrace{\lambda f.\lambda g.f\, g_1\, g}_{(e \to T) \to T}$$

This in turn implies something a little strange about the semantics of transitive verbs. They cannot be type $e \to e \to T$. Why not? Imagine you had a trace in object position. It wouldn't be able to combine with the verb!

$$[\![\mathsf{likes}]\!] = \underbrace{\lambda Q.\lambda y.Q(\lambda x.\mathbf{oldLikes}\, x\, y)}_{((e \to T) \to T) \to e \to T}$$

Indefinites

What kind of dynamic-semantics should we give to indefinites? Solve for ?:

$$\underbrace{?}_{[\![\text{a linguist}]\!]} + \underbrace{\lambda x.\lambda g.\{g \mid x \in \text{left}\}}_{[\![\text{left}]\!]} = \lambda g.\{g \mid \exists x \in \text{ling} : x \in \text{left}\}$$

Again, this fully determines $[\![\text{a linguist}]\!]$. What should it be?

Indefinites

What kind of dynamic-semantics should we give to indefinites? Solve for ?:

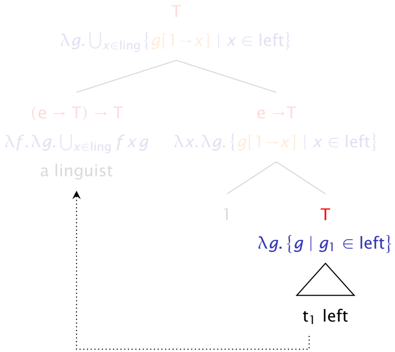
$$\underbrace{?}_{[\![\text{a linguist}]\!]} + \underbrace{\lambda x.\lambda g.\{g \mid x \in \text{left}\}}_{[\![\text{left}]\!]} = \lambda g.\{g \mid \exists x \in \text{ling} : x \in \text{left}\}$$

Again, this fully determines $[\![\text{a linguist}]\!]$. What should it be?

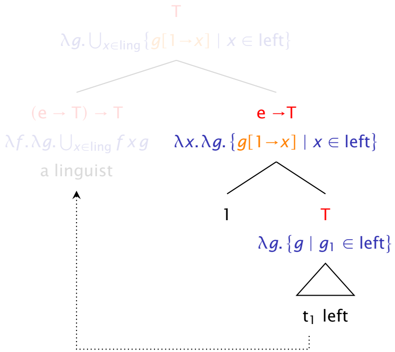
$$[\![\text{a linguist}]\!] = \underbrace{\lambda f.\lambda g.\bigcup_{x \in \text{ling}} f\,x\,g}_{(\text{e} \to \text{T}) \to \text{T}}$$

Then $[\![\text{a linguist left}]\!] = \lambda g.\bigcup_{x \in \text{ling}} \{g \mid x \in \text{left}\}$, which is either empty if no linguists left, or which is $\{g\}$ if some linguist left.

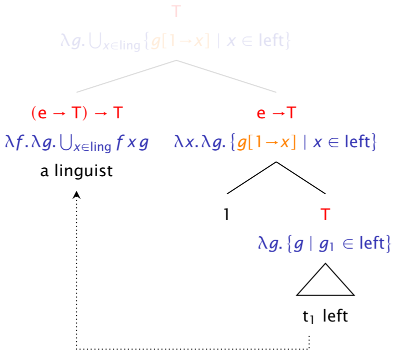
Example: *a linguist t left*



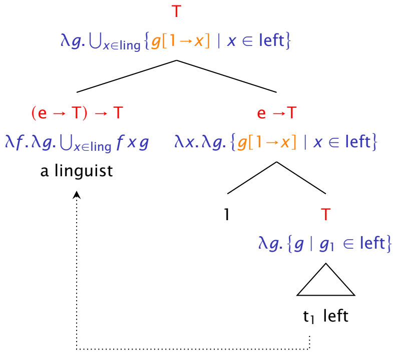
Example: *a linguist t left*



Example: *a linguist t left*



Example: *a linguist t left*



Input-output

$$g \dashrightarrow \text{Polly}^{n} \text{ left} \dashrightarrow g^{n\to \mathbf{p}}$$

$$g \dashrightarrow \text{a linguist}^{n} \text{ left} \dashrightarrow \left\{ g^{n\to \mathbf{a}},\ g^{n\to \mathbf{b}},\ g^{n\to \mathbf{c}},\ g^{n\to \mathbf{d}},\ g^{n\to \mathbf{e}} \right\}$$

Binding info, stored, can be passed:

$$[\![\text{and}]\!] = \underbrace{\lambda r.\lambda l.\lambda g.\bigcup_{h \in l g} r\, h}_{\mathrm{T}\to\mathrm{T}\to\mathrm{T}}$$

Going 'variable-free'

Exercise: restate compositional dynamics, using PLA! I'll get you started. Again, we'll keep the system relational, which is no loss since PLA, like FCS, is distributive:

$$\mathrm{T} ::= \underbrace{\mathrm{s} \to \{\mathrm{s}\}}_{\text{s is the type of sequences of entities}}$$

Then meanings for verbs can be stated in a way parallel to our previous system:

$$[\![\text{left}]\!] = \underbrace{\lambda x.\lambda e.\{e \mid x \in \text{left}\}}_{\mathrm{e} \to \mathrm{T}}$$

And predicate abstraction works by *extending the input sequence*:

$$[\![\uparrow \alpha]\!] = \underbrace{\lambda x.\lambda e.[\![\alpha]\!](e \cdot x)}_{\mathrm{e} \to \mathrm{T}}$$

Barwise, Jon. 1987. Noun phrases, generalized quantifiers, and anaphora. In Peter Gärdenfors (ed.), *Generalized Quantifiers*, 1–29. Dordrecht: Reidel. https://doi.org/10.1007/978-94-009-3381-1_1.

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

Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*. University of Massachusetts, Amherst Ph.D. thesis. http://semanticsarchive.net/Archive/Tk0ZmYyY/.

Heim, Irene. 1983. File change semantics and the familiarity theory of definiteness. In Rainer Bäuerle, Christoph Schwarze & Arnim von Stechow (eds.), *Meaning, use and interpretation of language*, 164–190. Berlin: Walter de Gruyter. https://doi.org/10.1515/9783110852820.164.

Muskens, Reinhard. 1996. Combining Montague semantics and discourse representation. *Linguistics and Philosophy* 19(2). 143–186. https://doi.org/10.1007/BF00635836.

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. https://doi.org/10.1007/978-94-009-3381-1_9.

Stalnaker, Robert. 1978. Assertion. In Peter Cole (ed.), *Pragmatics*, vol. 9 (Syntax and Semantics), 315–332. New York: Academic Press.