Classic Montagovian semantics deals with aboutness as primary, whereas dynamic semantics deals with information update as primary. What might a semantics look like that deals with interaction -- that is, agents seeking and providing information -- as primary?

I describe a model of interaction that I am working on for simple information exchanges. The model depicts interaction by graph rewrites and denotes interaction by message relations. It promises to formalize aboutness and information update on a new foundation, while remaining compatible with existing semantic tools like the lambda-calculus.

---

Interaction among
- people (information updates)
- things (aboutness)
- modules (interface)
- constituents (syntax)
- ...

---
(3) Rewriting rules capture the dynamics of
(a) computation

\[ \text{AND} \quad \Rightarrow \quad \text{No} \]

(b) communication

\[ \text{No} \quad \text{Bool} \quad \Rightarrow \quad \text{No} \quad \text{Bool} \]

(4) Denotations = "context semantics"; "game strategies"

\[ [\text{subgraph}] = \{ \langle \text{input, port, output, port} \rangle, \ldots \} \]

(Actually, easier to use a symmetric relation between message-port pairs.)

Port names are arbitrary
(a) \[ [\text{NO}]_{\text{BOOL}} \downarrow \pi_0 = \{ \langle ?, \pi_0, \text{NO}, \pi_0 \rangle \} \]

(b) \[ [\text{AND}]_{\text{BOOL}} \downarrow \pi_0 = \{ \langle ?, \pi_0, ?, \pi_1 \rangle, \langle \text{NO}, \pi_1, \text{NO}, \pi_0 \rangle, \langle \text{YES}, \pi_1, ?, \pi_2 \rangle, \langle \text{NO}, \pi_2, \text{NO}, \pi_0 \rangle, \langle \text{YES}, \pi_2, \text{YES}, \pi_0 \rangle \} \]

(c) Composition:

\[ \{ \langle M_0, \pi_0, \mu_0, \pi_n \rangle \mid \pi_0, \pi_n \text{ are external ports}, \]
\[ \pi_i, \ldots, \pi_{i-1} \text{ are internal ports,} \]
\[ \langle \mu_i, \pi_i, \mu_{i+1}, \pi_{i+1} \rangle \in \text{[[A]] U [[B]]} \]
\[ \text{(or its reverse) for } i=0, \ldots, n-1 \} \]

(d) Graph rewriting preserves denotation:

\[ [\text{AND}]_{\text{BOOL}} \downarrow \pi_0 = \{ \langle ?, \pi_0, \text{NO}, \pi_0 \rangle \} = [[\text{AND}]_{\text{BOOL}} \downarrow \pi_0] \]

(e) \[ [\pi' \leftarrow \pi] = \{ \langle \mu, \pi, \mu, \pi' \rangle \mid \mu \text{ is a message} \} \]
(6) a. \(\lambda x \cdot x\)  
\[\text{Graph}\]

b. \(\lambda x \cdot E\)
\[\text{Graph}\]

c. \(F(E)\)
\[\text{Graph}\]

(7) a. \(\lambda x \cdot \lambda y \cdot x\)
\[\text{Graph}\]

b. \(\lambda x \cdot \lambda c \cdot c(x)\)
\[\text{Consider only "affine" \(\lambda\)-terms for now — no variable appears multiple times. "Sharing" complicates the picture.}\]

c. \(\lambda x \cdot \lambda y \cdot x\)
\[\text{Graph}\]

d. \(\lambda x \cdot \lambda y \cdot y\)
\[\text{Graph}\]

(7c) is like NO  
(7d) is like YES  
(7e) is like AND.

\(\text{Graph}\)

e. \(\lambda f \cdot \lambda g \cdot f(\lambda x \cdot \lambda y \cdot x)(g)\)
\[\text{Graph}\]
(8) \( \lambda \)-conversion as a graph rewrite

(9) Example: \( \text{AND (NO) (YES)} \Rightarrow \text{NO} \)

(10) a. \[
\begin{array}{c}
\text{Continuation argument} \\
\text{function}
\end{array}
\] = \{ \langle \mu, \text{continuation}, \langle 0, \mu \rangle, \text{function} \rangle, \\
\langle \mu, \text{argument}, \langle 1, \mu \rangle, \text{function} \rangle \\
\text{for } \mu \text{ a message} \}

b. \[
\begin{array}{c}
\text{function} \\
\text{body} \\
\text{argument}
\end{array}
\] = \{ \langle \langle 0, \mu \rangle, \text{function}, \mu, \text{body} \rangle, \\
\langle \langle 1, \mu \rangle, \text{function}, \mu, \text{argument} \rangle \\
\text{for } \mu \text{ a message} \}