Monadic dynamic semantics: Side effects and scope

Simon Charlow [1] Dylan Bumford [2]

[1] Rutgers University [2] New York University

Fourth Workshop on Natural Language and Computer Science
July 10, 2016

Overview

- Old, extremely well-studied patterns concerning the **scope and binding** properties of **indefinites**
- A minimal semantic analysis, using **monads**
- Immediate integration into a compositional grammar, via **scope-taking**, and empirical benefits thereof
- Scope-taking lets different kind of effects interact **modularly**; the relative inflexibility of monads is no cause for concern

Where we are

Indefinites and discourse referents

Monadic dynamic semantics

Compositionality and scope

Things it does well

Modularity

What's special about indefinites

- An old chestnut: with respect to anaphora, indefinites have more in common with referential expressions than they do with quantifiers (Geach 1962; Evans 1980; Heim 1982, . . .).

(1) Cross-sentential anaphora:
{Polly, a linguist}$_i$ left. She$_i$ was tired.
*{No, every} linguist$_i$ left. She$_i$ was tired.

(2) Donkey anaphora:
Everyone who saw {Polly, a linguist}$_i$ waved to her$_i$.
*Everyone who saw {no, every} linguist$_i$ waved to her$_i$.

- Today: cross-sentential focus (but what we say extends to donkeys).

The puzzle: indefinites don't refer, right?

- To which individual does *a linguist* refer? None of em, really.
- Indeed, indefinites standardly typed as quantifiers:[1]

$$
\begin{aligned}
[\![\textit{Polly}]\!] &= \mathsf{p} && \text{type: } e \\
[\![\textit{a linguist}]\!] &= \lambda c.\, \exists x.\, \mathsf{ling}\, x \wedge c\, x && \text{type: } (e \to t) \to t \\
[\![\textit{every linguist}]\!] &= \lambda c.\, \forall x.\, \mathsf{ling}\, x \Rightarrow c\, x && \text{type: } (e \to t) \to t
\end{aligned}
$$

- But treating indefinites like quantifiers wrongly predicts they should pattern like quantifiers w.r.t. anaphora!

[1] e is a domain of *individuals* $\{\mathsf{polly}, \mathsf{bob}, \ldots\}$; t is a domain of *truth values* $\{\mathbb{T}, \mathbb{F}\}$

Where we are

Dynamic semantics

- In **dynamic semantics**, sentences encode *state transitions*, type $\gamma \to \{\gamma\}$ (e.g., Barwise 1987; Groenendijk & Stokhof 1991):

$$[\![\textit{Polly left}]\!] = \lambda i.\begin{cases}\{i + \mathrm{p}\} & \textbf{if } \mathrm{left}\,\mathrm{p} \\ \{\quad\} & \textbf{otherwise}\end{cases}$$

$$[\![\textit{she was tired}]\!] = \lambda i.\begin{cases}\{i\} & \textbf{if } \mathrm{tired}\, i_{\lightning} \\ \{\,\} & \textbf{otherwise}\end{cases}$$

- Sentences with indefinites encode *nondeterministic* state transitions:

$$[\![\textit{a linguist left}]\!] = \lambda i.\{i + x \mid \mathrm{ling}\, x, \mathrm{left}\, x\}$$

- Sentential concatenation is just *relation composition*:

$$[\![L\,;R]\!] = \lambda i.\bigcup_{j \in L\,i} R\,j$$

Pictorially

$$i \dashrightarrow [\![\,\textit{Polly left}\,]\!] \dashrightarrow i+\mathrm{p} \dashrightarrow [\![\textit{she was tired}]\!] \dashrightarrow i+\mathrm{p}$$

$$\begin{tikzcd}
 & & i+\mathrm{a} \arrow[r, dotted] & {[\![\textit{she was tired}]\!]} \arrow[r, dotted] & i+\mathrm{a} \\
 & & i+\mathrm{b} \arrow[r, dotted] & {[\![\textit{she was tired}]\!]} \arrow[r, dotted] & {} \\
i \arrow[r, dotted] & {[\![\textit{a ling left}]\!]} \arrow[uur, dotted] \arrow[ur, dotted] \arrow[r, dotted] \arrow[dr, dotted] & i+\mathrm{c} \arrow[r, dotted] & {[\![\textit{she was tired}]\!]} \arrow[r, dotted] & {} \\
 & & i+\mathrm{d} \arrow[r, dotted] & {[\![\textit{she was tired}]\!]} \arrow[r, dotted] & i+\mathrm{d}
\end{tikzcd}$$

A bit of metasemantics

- Sentences — things we associate with truth values or facts — are the only things it makes sense to associate with type $\gamma \to \{\gamma\}$.

$$\phi\, i = \{\ \} \Leftrightarrow \phi \text{ is } \textbf{false} \text{ at } i$$
$$\phi\, i \neq \{\ \} \Leftrightarrow \phi \text{ is } \textbf{true} \text{ at } i$$

- That means that in order to capture *sub-sentential* dynamic effects, all denotations will need to be “lifted” into higher-order functions that operate on sentence-sized constituents:

$$\begin{aligned}
[\![\mathit{Polly}]\!] &= \lambda ci.\, c\, \mathrm{p}\,(i + \mathrm{p}) && \text{type: } (e \to \gamma \to \{\gamma\}) \to \gamma \to \{\gamma\} \\
[\![\mathit{she}]\!] &= \lambda ci.\, c\, i_{\not z}\, i && \text{type: } (e \to \gamma \to \{\gamma\}) \to \gamma \to \{\gamma\} \\
[\![\mathit{a\ linguist}]\!] &= \lambda ci.\, \bigcup_{\mathrm{ling}\, x} c\, x\,(i + x) && \text{type: } (e \to \gamma \to \{\gamma\}) \to \gamma \to \{\gamma\}
\end{aligned}$$

Our view: dynamics via nondeterministic, tagged values

- Meaning for a proper name-containing sentence:

$$[\![\textit{Polly left}]\!] :: \gamma \to \{(t, \gamma)\}$$
$$[\![\textit{Polly left}]\!] = \lambda i.\, \{(\text{left}\, \text{p}, i + \text{p})\}$$

- Meaning for an indefinite-containing sentence:

$$[\![\textit{a linguist left}]\!] :: \gamma \to \{(t, \gamma)\}$$
$$[\![\textit{a linguist left}]\!] = \lambda i.\, \{(\text{left}\, x, i + x) \mid \text{ling}\, x\}$$

- Compared with the standard dynamic approach:
  - Old: returning an updated state, **conditional on some fact**
  - New: **unconditionally** pairing a fact with an updated state

Generalized to "referring" expressions

- Meaning for a proper name:

$$[\![\mathit{Polly}]\!] :: \gamma \to \{(e, \gamma)\}$$
$$[\![\mathit{Polly}]\!] = \lambda i. \{(\mathrm{p}, i + \mathrm{p})\}$$

- Meaning for an indefinite:

$$[\![\mathit{a\ linguist}]\!] :: \gamma \to \{(e, \gamma)\}$$
$$[\![\mathit{a\ linguist}]\!] = \lambda i. \{(x, i + x) \mid \mathrm{ling}\, x\}$$

- Compared with the standard dynamic approach:
  - Old: higher-order functions
  - New: pairing an individual with an updated state

Fully general: dynamic effects for any type

- A dynamic a, $\ulcorner \mathsf{D}a \urcorner$, has the following type:

$$\mathsf{D}a ::= \gamma \to \{(a, \gamma)\}$$

- Recasting our proposed meanings in terms of D:

$$\begin{aligned} [\![\textit{a linguist}]\!] &:: \mathsf{D}e \\ [\![\textit{a linguist left}]\!] &:: \mathsf{D}t \end{aligned}$$

Monads

- D is *monadic* (Moggi 1989; Wadler 1994, 1995; Shan 2002; Unger 2012; and many others), in that it has two functions η and $(\star)$:

$$\begin{array}{llll} \eta & :: a \to \mathrm{D}a & \qquad (\star) & :: \mathrm{D}a \to (a \to \mathrm{D}b) \to \mathrm{D}b \\ \eta\, x & := \lambda i.\, \{(x, i)\} & \qquad m \star c & := \lambda i. \bigcup_{(x,j) \in m\, i} c\, x\, j \end{array}$$

- η is an "injection" function, and $(\star)$ a recipe for plugging a Da into an $a \to \mathrm{D}b$ function to yield a Db
- η and $(\star)$ must satisfy certain properties, which needn't detain us, except for the crucial point that $(\star)$ is **associative**, in the following sense:

$$(m \star \lambda x.\, c\, x) \star k \equiv m \star (\lambda x.\, c\, x \star k)$$

Where we are

Indefinites and discourse referents

Monadic dynamic semantics

Compositionality and scope

Things it does well

Modularity

Composing meanings?

- Compositionality: how are the meanings of syntactically complex units built from the meanings of their parts?
- In this case: how should constituents that introduce or rely on dynamic effects combine with "normal" material?

$$\lambda i.\,\{(\mathsf{p}\,\mathsf{met}\,x,\, i + \mathsf{p} + x) \mid \mathsf{ling}\,x\}$$

???

Polly — ???

$\lambda i.\,\{(\mathsf{p}, i + \mathsf{p})\}$

met — a linguist

met — $\lambda i.\,\{(x, i + x) \mid \mathsf{ling}\,x\}$

Interlude: quantificational ambiguity

- This sentence has two readings (one quite implausible):

(3) An American flag flies in front of every embassy.
$\rightsquigarrow \exists \gg \forall$, $\forall \gg \exists$

- What kind of ambiguity? Doesn't seem lexical or structural.

Quantificational ambiguity as scope ambiguity

- Linguists since Montague (1974) locate this ambiguity in two possible **scopings** of *an American flag* and *every embassy*:

$$
\begin{array}{cccccccc}
\text{an American flag} & & \text{every embassy} & & [& t_x & \text{fifos} & t_y \ ] \\
\mathcal{F} & (\ \lambda x. & \mathcal{E} & (\ \lambda y. & & x & \text{fifos} & y\ )\ )
\end{array}
$$

$$
\begin{array}{cccccccc}
\text{every embassy} & & \text{an American flag} & & [& t_x & \text{fifos} & t_y \ ] \\
\mathcal{E} & (\ \lambda y. & \mathcal{F} & (\ \lambda x. & & x & \text{fifos} & y\ )\ )
\end{array}
$$

- To **take scope over** E is to have E contained within your argument.
- Many approaches on the books (syntactic, logical, continuations). Choice immaterial, though we're naturally inclined towards continuations-based analyses (Barker 2002; Barker & Shan 2014; Charlow 2014).

Scope-taking (by any means) feeds η and $\star$

Classical:

$$\begin{array}{ccccccccc} \text{a linguist} & & [& \text{John} & \text{met} & t_x &] & \\ (\lambda c.\exists x.\,\text{ling}\,x \wedge c\,x) & (\;\lambda x. & (& \text{j} & \text{met} & x &) &) \end{array}$$

Dynamic:

$$\begin{array}{cccccccccc} \text{a linguist} & & & & [& \text{John} & \text{met} & t_x &] & \\ (\lambda i.\{(x, i + x) \mid \text{ling}\,x\}) & \star & (\;\lambda x. & \eta & (& \text{j} & \text{met} & x &) &) \end{array}$$

- η and $(\star)$, together with any mechanism for scope-taking, provide the glue to thread effect-ful meanings together.

Scope-taking (by any means) feeds η and $\star$

Classical:

$$\exists x.\, \mathsf{ling}\, x \wedge \mathsf{j}\, \mathsf{met}\, x$$

$$\begin{array}{ccccccc} \text{a linguist} & & [& \text{John} & \text{met} & t_x &] \\ (\lambda c.\, \exists x.\, \mathsf{ling}\, x \wedge c\, x) & (\ \lambda x. & (& \mathsf{j} & \mathsf{met} & x &)\ ) \end{array}$$

Dynamic:

$$\lambda i.\, \{(\mathsf{j}\, \mathsf{met}\, x, i + x) \mid \mathsf{ling}\, x\}$$

$$\begin{array}{ccccccc} \text{a linguist} & & [& \text{John} & \text{met} & t_x &] \\ (\lambda i.\, \{(x, i + x) \mid \mathsf{ling}\, x\}) \ \star & (\ \lambda x.\ \eta & (& \mathsf{j} & \mathsf{met} & x &)\ ) \end{array}$$

- η and $(\star)$, together with any mechanism for scope-taking, provide the glue to thread effect-ful meanings together.

Example derivations

$$\begin{array}{ccccccccccc}
\text{John} & & & & [& \text{Mary} & \text{gave} & t_x & \text{hope} &] \\
\lambda i.\,\{(\mathrm{j}, i+\mathrm{j})\} & \star & \lambda x. & \eta & (& \mathrm{m} & \mathrm{gave} & x & \mathrm{h} &)
\end{array}$$

$$\begin{array}{cccccccccccc}
\text{John} & & & \text{his book} & & & & [& \text{Mary} & \text{gave} & t_x & t_y &] \\
\lambda i.\,\{(\mathrm{j}, i+\mathrm{j})\} & \star & \lambda x. & \lambda i.\,\{(\mathrm{bk}\, i_{\not{z}}, i)\} & \star & \lambda y. & \eta & (& \mathrm{m} & \mathrm{gave} & x & y &)
\end{array}$$

$$\begin{array}{cccccccccccc}
\text{Someone} & & & \text{his book} & & & & [& \text{Mary} & \text{gave} & t_x & t_y &] \\
\lambda i.\,\{(x, i+x) \mid x_e\} & \star & \lambda x. & \lambda i.\,\{(\mathrm{bk}\, i_{\not{z}}, i)\} & \star & \lambda y. & \eta & (& \mathrm{m} & \mathrm{gave} & x & y &)
\end{array}$$

Example derivations

$$\lambda i.\, \{(\mathrm{m\,gave\,j\,h},\, i + \mathrm{j})\}$$

$$\begin{array}{cccccccccc}
\text{John} & & & & [& \text{Mary} & \text{gave} & t_x & \text{hope} &] \\
\lambda i.\, \{(\mathrm{j},\, i + \mathrm{j})\} & \star & \lambda x. & \eta & (& \mathrm{m} & \mathrm{gave} & x & \mathrm{h} &)
\end{array}$$

$$\begin{array}{ccccccccccccc}
\text{John} & & & \text{his book} & & & & [& \text{Mary} & \text{gave} & t_x & t_y &] \\
\lambda i.\, \{(\mathrm{j},\, i + \mathrm{j})\} & \star & \lambda x. & \lambda i.\, \{(\mathrm{bk}\, i_{\lightning},\, i)\} & \star & \lambda y. & \eta & (& \mathrm{m} & \mathrm{gave} & x & y &)
\end{array}$$

$$\begin{array}{ccccccccccccc}
\text{Someone} & & & \text{his book} & & & & [& \text{Mary} & \text{gave} & t_x & t_y &] \\
\lambda i.\, \{(x,\, i + x) \mid x_e\} & \star & \lambda x. & \lambda i.\, \{(\mathrm{bk}\, i_{\lightning},\, i)\} & \star & \lambda y. & \eta & (& \mathrm{m} & \mathrm{gave} & x & y &)
\end{array}$$

Example derivations

$$\lambda i.\,\{(\mathrm{m\,gave\,j\,h},\, i+\mathrm{j})\}$$

$$\begin{array}{ccccccccc}
\text{John} & & & [& \text{Mary} & \text{gave} & t_x & \text{hope} &] \\
\lambda i.\,\{(\mathrm{j},\, i+\mathrm{j})\} & \star & \lambda x.\ \eta & (& \mathrm{m} & \mathrm{gave} & x & \mathrm{h} &)
\end{array}$$

$$\lambda i.\,\{(\mathrm{m\,gave\,j\,(bk\,j)},\, i+\mathrm{j})\}$$

$$\begin{array}{ccccccccc}
\text{John} & & \text{his book} & & [& \text{Mary} & \text{gave} & t_x & t_y\ ] \\
\lambda i.\,\{(\mathrm{j},\, i+\mathrm{j})\} & \star & \lambda x.\ \lambda i.\,\{(\mathrm{bk}\,i_{\lightning},\, i)\} & \star & \lambda y.\ \eta\ (& \mathrm{m} & \mathrm{gave} & x & y\ )
\end{array}$$

$$\begin{array}{ccccccccc}
\text{Someone} & & \text{his book} & & [& \text{Mary} & \text{gave} & t_x & t_y\ ] \\
\lambda i.\,\{(x,\, i+x) \mid x_e\} & \star & \lambda x.\ \lambda i.\,\{(\mathrm{bk}\,i_{\lightning},\, i)\} & \star & \lambda y.\ \eta\ (& \mathrm{m} & \mathrm{gave} & x & y\ )
\end{array}$$

Example derivations

$$\lambda i.\,\{(\mathrm{m\,gave\,j\,h},\, i+\mathrm{j})\}$$

$$\begin{array}{ccccccccccc}
\text{John} & & & & [& \text{Mary} & \text{gave} & t_x & \text{hope} &] \\
\lambda i.\,\{(\mathrm{j},\, i+\mathrm{j})\} & \star & \lambda x. & \eta & (& \mathrm{m} & \mathrm{gave} & x & \mathrm{h} &)
\end{array}$$

$$\lambda i.\,\{(\mathrm{m\,gave\,j\,(bk\,j)},\, i+\mathrm{j})\}$$

$$\begin{array}{cccccccccccc}
\text{John} & & & \text{his book} & & & & [& \text{Mary} & \text{gave} & t_x & t_y &] \\
\lambda i.\,\{(\mathrm{j},\, i+\mathrm{j})\} & \star & \lambda x. & \lambda i.\,\{(\mathrm{bk}\,i_{\lightning},\, i)\} & \star & \lambda y. & \eta & (& \mathrm{m} & \mathrm{gave} & x & y &)
\end{array}$$

$$\lambda i.\,\{(\mathrm{m\,gave}\,x\,(\mathrm{bk}\,x),\, i+x) \mid x_e\}$$

$$\begin{array}{cccccccccccc}
\text{Someone} & & & \text{his book} & & & & [& \text{Mary} & \text{gave} & t_x & t_y &] \\
\lambda i.\,\{(x,\, i+x) \mid x_e\} & \star & \lambda x. & \lambda i.\,\{(\mathrm{bk}\,i_{\lightning},\, i)\} & \star & \lambda y. & \eta & (& \mathrm{m} & \mathrm{gave} & x & y &)
\end{array}$$

Note on do-notation

- Haskell programmers write code that looks like this:

$$\begin{aligned}&\text{do } x \leftarrow m\\&\quad\;\; y \leftarrow n\\&\quad\;\; \texttt{return}\,(f\,x\,y)\end{aligned}$$

- ...Which is a sugaring of this:

$$m \star \lambda x.\, n \star \lambda y.\, \eta\,(f\,x\,y)$$

- ...Which, interestingly, has a rather direct correspondence with the scoped logical forms we make use of here (cf. Wadler 1994):

$$m \;\star\; \lambda x.\; n \;\star\; \lambda y.\; \eta\;(\;f\;\;x\;\;y\;)$$

Where we are

Indefinites and discourse referents

Monadic dynamic semantics

Compositionality and scope

Things it does well

Modularity

Cross-sentential anaphora

John left he whistled [t_p and t_q]

$$(\mathbf{jn} \star \lambda x.\,\eta\,(\text{left}\,x)) \;\star\; \lambda p.\;(\mathbf{he} \star \lambda y.\,\eta\,(\text{whis}\,y)) \;\star\; \lambda q.\;\eta\;(\;p \;\wedge\; q\;)$$

Someone left he whistled [t_p and t_q]

$$(\mathbf{so} \star \lambda x.\,\eta\,(\text{left}\,x)) \;\star\; \lambda p.\;(\mathbf{he} \star \lambda y.\,\eta\,(\text{whis}\,y)) \;\star\; \lambda q.\;\eta\;(\;p \;\wedge\; q\;)$$

Cross-sentential anaphora

John left   he whistled   [t_p and t_q]

$$\mathbf{jn} \star \lambda x.\, \eta\,(\text{left}\, x) \quad \star \quad \lambda p.\quad \mathbf{he} \star \lambda y.\, \eta\,(\text{whis}\, y) \quad \star \quad \lambda q.\quad \eta \;(\; p \quad \wedge \quad q \;)$$

Someone left   he whistled   [t_p and t_q]

$$\mathbf{so} \star \lambda x.\, \eta\,(\text{left}\, x) \quad \star \quad \lambda p.\quad \mathbf{he} \star \lambda y.\, \eta\,(\text{whis}\, y) \quad \star \quad \lambda q.\quad \eta \;(\; p \quad \wedge \quad q \;)$$

Cross-sentential anaphora

$$\lambda i.\, \{(\mathrm{left}\, \mathrm{j} \wedge \mathrm{whis}\, \mathrm{j},\, i + \mathrm{j})\}$$

John left he whistled [t_p and t_q]

$$\mathbf{jn} \star \lambda x.\, \eta\, (\mathrm{left}\, x) \quad \star \quad \lambda p. \quad \mathbf{he} \star \lambda y.\, \eta\, (\mathrm{whis}\, y) \quad \star \quad \lambda q. \quad \eta \; (\; p \; \wedge \; q \;)$$

$$\lambda i.\, \{(\mathrm{left}\, x \wedge \mathrm{whis}\, x,\, i + x) \mid x_e\}$$

Someone left he whistled [t_p and t_q]

$$\mathbf{so} \star \lambda x.\, \eta\, (\mathrm{left}\, x) \quad \star \quad \lambda p. \quad \mathbf{he} \star \lambda y.\, \eta\, (\mathrm{whis}\, y) \quad \star \quad \lambda q. \quad \eta \; (\; p \; \wedge \; q \;)$$

Cross-sentential anaphora

$$\lambda i.\,\{(\mathrm{left}\,\mathrm{j} \wedge \mathrm{whis}\,\mathrm{j},\, i + \mathrm{j})\}$$

John left   he whistled   [t_p and t_q]

$$\mathbf{jn} \star \lambda x.\,\eta\,(\mathrm{left}\,x) \quad \star \quad \lambda p.\ \ \mathbf{he} \star \lambda y.\,\eta\,(\mathrm{whis}\,y) \quad \star \quad \lambda q.\ \ \eta\ (\ p \ \ \wedge \ \ q\ )$$

$$\lambda i.\,\{(\mathrm{left}\,x \wedge \mathrm{whis}\,x,\, i + x) \mid x_e\}$$

Someone left   he whistled   [t_p and t_q]

$$\mathbf{so} \star \lambda x.\,\eta\,(\mathrm{left}\,x) \quad \star \quad \lambda p.\ \ \mathbf{he} \star \lambda y.\,\eta\,(\mathrm{whis}\,y) \quad \star \quad \lambda q.\ \ \eta\ (\ p \ \ \wedge \ \ q\ )$$

- Precisely mirrors the patterns with sub-clausal binding
- All the action is in $(\star)$; conjunction is classical

“Exceptional” wide scope

- Indefinites seem to have greater upward “scopal mobility” than true quantifiers (e.g., Fodor & Sag 1982):

(4) If a (certain) linguist shows up, it’ll be bedlam. $\exists \gg \Rightarrow$

(5) If every linguist shows up, it’ll be bedlam. $^*\forall \gg \Rightarrow$

- A direct consequence of the way nondeterminism persists through $(\star)$. Indeed, the account is parallel to cross-sentential anaphora!

$$\begin{array}{ccccccccc} \text{A ling shows up} & & & & [\ \text{if} & t_p, & \text{then} & \text{bedlam} &] \\ (\,\textbf{a-ling} \star \lambda x.\,\eta\,(\text{show-up}\,x)\,) & \star & \lambda p. & \eta & (& p & \Rightarrow & \mathsf{b} &) \end{array}$$

$$\begin{array}{ccccccccc} \text{Every ling shows up} & & & & [\ \text{if} & t_p, & \text{then} & \text{bedlam} &] \\ (\,\textbf{every-ling}\,(\lambda x.\,\text{show-up}\,x)\,) & \star & \lambda p. & \eta & (& p & \Rightarrow & \mathsf{b} &) \end{array}$$

“Exceptional” wide scope

- Indefinites seem to have greater upward “scopal mobility” than true quantifiers (e.g., Fodor & Sag 1982):

(4) If a (certain) linguist shows up, it’ll be bedlam. $\exists \gg \Rightarrow$

(5) If every linguist shows up, it’ll be bedlam. $^*\forall \gg \Rightarrow$

- A direct consequence of the way nondeterminism persists through $(\star)$. Indeed, the account is parallel to cross-sentential anaphora!

$$\begin{array}{cccccccccc} \text{A ling shows up} & & & & [& \text{if} & t_p, & \text{then} & \text{bedlam} &] \\ \textbf{a-ling} \star \lambda x.\, \eta\,(\text{show-up}\, x) & \star & \lambda p. & \eta & (& & p & \Rightarrow & \text{b} &) \end{array}$$

$$\begin{array}{cccccccccc} \text{Every ling shows up} & & & & [& \text{if} & t_p, & \text{then} & \text{bedlam} &] \\ (\,\textbf{every-ling}\,(\lambda x.\, \text{show-up}\, x)\,) & \star & \lambda p. & \eta & (& & p & \Rightarrow & \text{b} &) \end{array}$$

More generally

- Both cross-sentential anaphora and exceptional wide scope turn on the associativity of $(\star)$:

$$(m \star \lambda x.\, cx) \star k \equiv m \star (\lambda x.\, cx \star k)$$

- Though m's scope is confined to $(m \star \lambda x.\, cx)$ on the left, the result is equivalent to m having scope over k.

- This "action at a distance" — m influencing k even as m does not directly interact with k — is linguists' **island-insensitivity**.
  - An indefinite {provides an antecedent for a pronoun, nondeterministically infects a conditional}, even as the indefinite is evaluated inside a separate, smaller domain (its minimal tensed clause).

Where we are

Indefinites and discourse referents

Monadic dynamic semantics

Compositionality and scope

Things it does well

Modularity

Effects everywhere, island-insensitivity everywhere

- Monadic techniques useful for a broad range of effectful fragments of natural language:
  - Prosodic prominence/focus (Shan 2002; Charlow 2014)
  - Supplemental content (Giorgolo & Asudeh 2012)
  - Environment-sensitivity (Shan 2002; Ben-Avi & Winter 2007)
  - Presupposition/exception handling (Wadler 1995)
  - “Pure” nondeterminism (Charlow 2014)
- All predicted to — and do — show the same patterns of island-insensitivity

Effects are separable

- Non-dynamic effects abound in natural language.
- Yet one often hears worries that monads aren't closed under composition.
- It's not clear this should cause linguists to lose sleep:
  - Importantly, scope-taking guarantees that different kinds of effects *can* steer clear of one another.
  - In the present case, this ensures the interoperability of dynamic theorizing with the rest of semantics.
  - In short, effects perfectly well combined by not combining them!

Test case #1: focus

- Prosodic prominence ('focus') standardly analyzed as invoking a set of alternative utterances (Rooth 1985): $[\![\mathit{JOHN}]\!] = (\mathsf{j}, \mathsf{alts}\,\mathsf{j})$.
- Can be seen as an enriched, monadic type (Shan 2002):

$$
\begin{aligned}
\mathsf{P}a &::= (a, \{a\}) \\
\eta\, x &:= (x, \{x\}) \\
(x, ys) \star c &:= (\mathsf{fst}\,(c\,x), \bigcup_{y \in ys} \mathsf{snd}\,(c\,y))
\end{aligned}
$$

- Interacting with the dynamic bits *just works* (other layering possible!):

$$
\begin{array}{cccccccccccccc}
\text{JOHN} & & & & & \text{someone} & & & & [& t_x & \text{invited} & t_y &] & \\
\textbf{JN} & \star_{\mathsf{P}} & \lambda x. & \eta_{\mathsf{P}} & (& \textbf{so} & \star_{\mathsf{D}} & \lambda y. & \eta_{\mathsf{D}} & (& x & \text{inv} & y &) &)
\end{array}
$$

Test case #2: effects feed effects

- What's more, effects interact (in monadically predictable ways)

(6) John, who met a linguist$_i$, said she$_i$ was nice.

(7) A linguist$_i$ met John, who said she$_i$ was nice.

- The monad for supplemental content (cf. Giorgolo & Asudeh 2012), works by accumulating supplements qua conjuncts in a second dimension:

$$\begin{aligned} \mathsf{S}\,a &::= (a, t) \\ \eta\, x &:= (x, \mathbb{T}) \\ (x, p) \star c &:= (\mathsf{fst}\,(c\,x), p \wedge \mathsf{snd}\,(c\,x)) \end{aligned}$$

- Throw η_{S} and $(\star_{\mathsf{S}})$ into a dynamic grammar with η_{D} and $(\star_{\mathsf{D}})$, add some lexical entries for non-restrictive relativization, and stir. You're done.

Other constructs

- Not every "enriched type" gives rise to a monad
- Monad $\subset$ Applicative $\subset$ Functor
- Does what we say hold for "mere" functors and applicatives?

Yas

- Every functor F (ergo, every applicative, every monad) has a ‘mapping’ operation $(\circ)$, with the following type:

$$(\circ) :: (a \to b) \to \mathsf{F}a \to \mathsf{F}b$$

- Let’s flip it:

$$(\bullet) :: \mathsf{F}a \to \underbrace{(a \to b) \to \mathsf{F}b}_{\text{scope-taker}}$$

- $(\bullet)$ bears a striking resemblance to the monadic $(\star)$:

$$(\star) :: \mathsf{M}a \to \underbrace{(a \to \mathsf{M}b) \to \mathsf{M}b}_{\text{scope-taker}}$$

- Thus, $(\bullet)$ and scope can also be used to grease the compositional skids

Associativity

- For any functor, the following holds of its $(\bullet)$:

$$(f \bullet \lambda x.\, c\, x) \bullet k \equiv f \bullet (\lambda x.\, k\,(c\, x))$$

- This is a kind of associativity. Ergo, island-insensitivity — f affecting k at a distance — predicted!

Wrapping up

- We've sketched a monadic interface encapsulating hoary dynamic notions of natural language meaning. . .
  - Generates new empirical predictions (in particular, island-insensitivity)
  - Plugs into any existing grammar, with or without extant side effects, interacting as needed (or not) with other semantically rich linguistic bits
- Are there any linguistically attested interactions of effects that are beyond the expressive power of the scope/η/$\star$ mechanism?
- To what extent is this technique compatible with (or recapitulating) alternative effect-handling regimes?

References

Barker, Chris. 2002. Continuations and the nature of quantification. *Natural Language Semantics* 10(3). 211–242. http://dx.doi.org/10.1023/A:1022183511876.

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and natural language*. Oxford: Oxford University Press. http://dx.doi.org/10.1093/acprof:oso/9780199575015.001.0001.

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

Ben-Avi, Gilad & Yoad Winter. 2007. The semantics of intensionalization. In Reinhard Muskens (ed.), *Proceedings of the workshop on New Directions in Type-theoretic Grammars*, 98–112.

Charlow, Simon. 2014. *On the semantics of exceptional scope*: New York University Ph.D. thesis. http://semanticsarchive.net/Archive/2JmMWRjY/.

Evans, Gareth. 1980. Pronouns. *Linguistic Inquiry* 11(2). 337–362. https://www.jstor.org/stable/4178164.

Fodor, Janet Dean & Ivan A. Sag. 1982. Referential and quantificational indefinites. *Linguistics and Philosophy* 5(3). 355–398. http://dx.doi.org/10.1007/BF00351459.

Geach, Peter. 1962. *Reference and Generality*. Ithaca, NY: Cornell University Press.

Giorgolo, Gianluca & Ash Asudeh. 2012. $\langle M, \eta, \star \rangle$: Monads for conventional implicatures. In Ana Aguilar Guevara, Anna Chernilovskaya & Rick Nouwen (eds.), *Proceedings of Sinn und Bedeutung 16*, 265–278. MIT Working Papers in Linguistics. http://mitwpl.mit.edu/open/sub16/Giorgolo.pdf.

References (cont.)

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

Moggi, Eugenio. 1989. Computational lambda-calculus and monads. In *Proceedings of the Fourth Annual Symposium on Logic in computer science*, 14–23. Piscataway, NJ, USA: IEEE Press.

Montague, Richard. 1974. The proper treatment of quantification in ordinary English. In Richmond Thomason (ed.), *Formal Philosophy*, chap. 8, 247–270. New Haven: Yale University Press.

Rooth, Mats. 1985. *Association with focus*: University of Massachusetts, Amherst Ph.D. thesis.

Shan, Chung-chieh. 2002. Monads for natural language semantics. In Kristina Striegnitz (ed.), *Proceedings of the ESSLLI 2001 Student Session*, 285–298.

Unger, Christina. 2012. Dynamic semantics as monadic computation. In Manabu Okumura, Daisuke Bekki & Ken Satoh (eds.), *New Frontiers in Artificial Intelligence JSAI-isAI 2011*, vol. 7258 Lecture Notes in Artificial Intelligence, 68–81. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-32090-3_7.

Wadler, Philip. 1994. Monads and composable continuations. *Lisp and Symbolic Computation* 7(1). 39–56. http://dx.doi.org/10.1007/BF01019944.

Wadler, Philip. 1995. Monads for functional programming. In Johan Jeuring & Erik Meijer (eds.), *Advanced Functional Programming*, vol. 925 Lecture Notes in Computer Science, 24–52. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/3-540-59451-5_2.