Monadic dynamic semantics for anaphora

Simon Charlow

Rutgers, The State University of New Jersey

OSU Dynamics Workshop · October 24, 2015

Goals for today

- I'll sketch a **monadic** dynamic semantics for discourse (and donkey) anaphora.
  - Dynamic semantics is **state** and **nondeterminism**.
  - A monadic dynamic semantics takes state and nondeterminism to be linguistic **side effects** (Shan 2002, 2005).
- Show why we should prefer this kind of approach to standard varieties of dynamic semantics:
  - Embodies more conservative view of lexical semantics.
  - Predicts wide variety of exceptional scope phenomena.
  - Super modular.
- Monadic dynamics suggests a fundamental connection between static alternatives-based and dynamic approaches to indefinites.

Where we are

Dynamic semantics

Monads

Monadic dynamic semantics

Features of the monadic account

Modularity

Basic data

- A familiar data point: Indefinites behave more like names than quantifiers with respect to anaphoric phenomena.

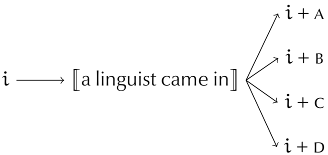
(1) {Polly$_i$, a linguist$_i$, *no linguist$_i$} came in. She$_i$ sat.

Dynamics (e.g., Groenendijk & Stokhof 1991; Dekker 1994)

- In a nutshell: sentences add **discourse referents** (drefs) to the "conversational scoreboard". E.g., for proper names:

$$i \longrightarrow [\![\text{Polly came in}]\!] \longrightarrow i + \text{P}$$

- Indefinites introduce drefs **nondeterministically**. E.g., if four linguists came in — A, B, C, and D — we'll have:



- Formally captured by modeling meanings as **relations on states**. E.g., here is a dynamic meaning for *a linguist came in*:

$$\lambda i.\ \{i + x \mid \text{LING}\ x \wedge \text{CAME}\ x\}$$

Going Montagovian

- Proper names:

$$\textbf{POLLY} \coloneqq \lambda \kappa i.\, \kappa\, \mathrm{P}\, (i + \mathrm{P})$$

- Indefinites:

$$\textbf{A.LING} \coloneqq \lambda \kappa i.\, \bigcup_{\mathrm{LING}\, x} \kappa\, x\, (i + x)$$

- Pronouns:

$$\textbf{SHE}_0 \coloneqq \lambda \kappa i.\, \kappa\, i_0\, i$$

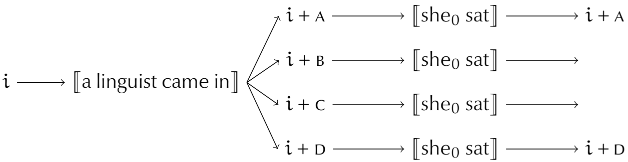
- Things like VPs will denote functions from individuals into dynamic propositions (i.e. relations on states). Meaning composition is therefore simple functional application.

Dynamic conjunction

- Given relational sentence meanings, sentential conjunction amounts to relation composition:

$$\textsc{and} \coloneqq \lambda R L i. \bigcup_{j \in L i} R\, j$$

- Deriving *a linguist came in, (and) she sat*:



- Given as a relation on states:

$$\lambda i.\, \{ i + x \mid \textsc{ling}\, x \wedge \textsc{came}\, x \wedge \textsc{sat}\, x \}$$

- Downstream indefinites may create further branching.

Getting closure

- Dynamic binding isn't anything-goes:

  (2) I don't own a radio. #It's a Panasonic.

  (3) Every boy fed a donkey. #It's braying. $(\forall > \exists)$

- Negation is externally static (i.e., closed):

$$\textbf{NOT} = \lambda S i. \begin{cases} \{i\} \text{ if } S\, i = \{\ \} \\ \{\ \} \text{ otherwise} \end{cases}$$

- Quantifiers, too:

$$\textbf{EVERY.BOY} = \lambda \kappa i. \begin{cases} \{i\} \text{ if } \forall x \in \textsc{boy}.\, \kappa\, x\, i \neq \{\ \} \\ \{\ \} \text{ otherwise} \end{cases}$$

Where we are

Dynamic semantics

Monads

Monadic dynamic semantics

Features of the monadic account

Modularity

What are monads?

- Construct from category theory and computer science used to talk about **side effects** (roughly, fancy things that happen in computations besides application of functions to values).
  - Some key citations: Moggi 1989; Wadler 1992, 1994, 1995; Liang et al. 1995; Shan 2002; Giorgolo & Asudeh 2012; Unger 2012.
- Gives a unified perspective on how meanings inhabiting "fancy" types, abbreviated $M\alpha$, interact with more quotidian bits.

This section

- Introducing you to two monads and how they relate to extant modes of composition in the semantics literature:
  - **Reader** monad: index-dependence
  - **Set** monad: nondeterminism
- As linguists, we can think of a monadic semantics as contributing two combinators or type-shifters to the grammar, $\boxdot$ and $\star$:
  - $\boxdot$ lifts boring things into maximally boring fancy things
  - $\star$ tells us how to combine fancy things
- As we'll see, **scope-taking** is an essential part of the story.

Example #1: Reader monad

- Task: compositionally integrating index-sensitive meanings:

$$\textsc{she}_0 \coloneqq \lambda i.\, i_0$$

- Usual approach is enriching the semantics of combination (e.g., Heim & Kratzer 1998):

$$[\![X\,Y]\!]^i = [\![X]\!]^i\,[\![Y]\!]^i$$

- In the monadic setting, the two combinators look like so:

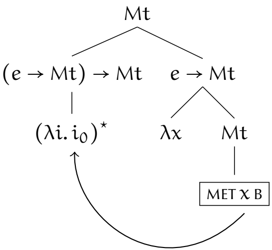
$$\boxed{x} \coloneqq \lambda i.\, x \qquad\qquad m^\star \coloneqq \lambda \kappa i.\, \kappa\,(m\,i)\,i$$

- A fancy a in the Reader monad, '$\mathsf{M}a$', is an index-dependent a:

$$\mathsf{M}a ::= i \to a$$

Reader monad derivation

- An example of how this works for *Bob met her$_0$*:



- Result: $\lambda i.\ \text{MET}\ i_0\ \text{B}$. (Same as what Heim & Kratzer derive.)
- This pattern will be repeated time and again. The fancy thing takes scope via ⋆, and $\boxed{\cdot}$ applies to its remnant.

Example #2: Set monad

- It is sometimes useful to entertain multiple values in parallel (e.g., Hamblin 1973; Kratzer & Shimoyama 2002):

$$[\![\text{a linguist}]\!] = \{x \mid \textsc{ling}\, x\}$$
$$[\![\text{Bob met a linguist}]\!] = \{\textsc{met}\, x\, \textsc{b} \mid \textsc{ling}\, x\}$$

- Usual approach is to enrich composition to handle sets:

$$[\![\text{A B}]\!] = \{f\, x \mid f \in [\![\text{A}]\!] \wedge x \in [\![\text{B}]\!]\}$$

- In the monadic setting, the two combinators look like so:

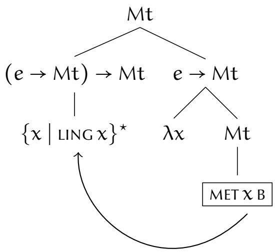
$$\boxed{x} \coloneqq \{x\} \qquad\qquad m^\star \coloneqq \lambda \kappa. \bigcup_{x \in m} \kappa\, x$$

- Emodies a notion of **nondeterministic** computation, where fancy things introduce alternatives into the semantics:

$$\mathrm{M}a ::= \{a\} \ \ (\text{i.e., } a \to t)$$

Set monad derivation

- How this works for *Bob met a linguist* (Charlow 2015):



- Gives the expected set of propositions, about different linguists:

$$\{\text{MET}\ x\ \text{B} \mid \text{LING}\ x\}$$

- Again, *exactly* the same pattern as Reader and State.

Monads, summed up

- Typing judgments, where $\mathsf{M}\mathfrak{a}$ should be read as "a fancy $\mathfrak{a}$"

$$\boxed{\cdot} :: \mathfrak{a} \to \mathsf{M}\mathfrak{a} \qquad \star :: \mathsf{M}\mathfrak{a} \to (\mathfrak{a} \to \mathsf{M}\mathfrak{b}) \to \mathsf{M}\mathfrak{b}$$

- Sub-cases:
  - Reader. $\mathsf{M}\mathfrak{a} ::= \mathfrak{i} \to \mathfrak{a}$
  - Set. $\mathsf{M}\mathfrak{a} ::= \{\mathfrak{a}\}$
- For any monad, $\boxed{x}^{\star} = \lambda\kappa.\, \kappa\, x$. Each monad thus implicates a different **decomposition** of LIFT (Partee 1986).

Compositionality

- The theory:
  - Find evidence for some side effects.
  - Posit some lexical items exploiting these side effects.
  - Fix the appropriate monad (i.e., a pair of $\boxed{\cdot}$ and $\star$).
  - Use $\boxed{\cdot}$, $\star$, and scope-taking (already present in your theory, I hope) to interface between the boring things and the fancy things.
- Plug in your favorite account of scope-taking. I'm using 'LFs', but your favorite account of scope will work just as well.
  - Proof-theoretic accounts (e.g., TLG).
  - Continuations + CCG (e.g., Shan & Barker 2006; Charlow 2014).
  - ...

Where we are

Dynamic semantics

Monads

Monadic dynamic semantics

Features of the monadic account

Modularity

Set the stage

- Dynamics relies on State, the ability to update indices, and nondeterminism (indefinites output *alternative* assignments).
- It's straightforward to fold dynamics into the monadic perspective.

State monad

- A generalization of the Reader monad allows meanings that **store**, as well as extract, anaphoric information (e.g., Unger 2012):

$$\textbf{POLLY} \coloneqq \lambda i.\, \langle \text{P}, i + \text{P} \rangle \qquad \textbf{SHE}_{\mathbf{0}} \coloneqq \lambda i.\, \langle i_0, i \rangle$$

- Here, the fancy types are functions from indices to pairs of values, and possibly-updated indices:

$$\text{M}a ::= i \to \langle a, i \rangle$$

- Monadic combinators again essentially follow from the types ($\langle x, y \rangle_l = x$, and $\langle x, y \rangle_r = y$):

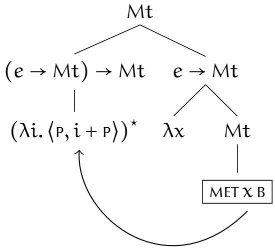
$$\boxed{x} \coloneqq \lambda i.\, \langle x, i \rangle \qquad m^\star \coloneqq \lambda \kappa i.\, \kappa\, (m\, i)_l\, (m\, i)_r$$

- Compare Reader:

$$\boxed{x} \coloneqq \lambda i.\, x \qquad m^\star \coloneqq \lambda \kappa i.\, \kappa\, (m\, i)\, i$$

State monad derivation

- An example of how this works for *Bob met Polly*:



- The result: $\lambda i.\, \langle \textsc{met p b}, i + \textsc{p} \rangle$.
- Along similar lines, we can derive a meaning for *she waved*:

$$\textbf{she}_0{}^{\star}\,(\lambda x.\, \boxed{\textsc{waved}\ x}\,) = \lambda i.\, \langle \textsc{waved}\ i_0, i \rangle$$

- How to bind pronouns? We'll see.

Adding nondeterminism to State

- One way to think of this is in terms of a new "fancy" type:

$$\mathrm{M}a ::= i \to \{\langle a, i \rangle\}$$

- The monadic operations essentially follow from the types:

$$\boxed{x} \coloneqq \lambda i.\,\{\langle x, i \rangle\} \qquad\qquad m^{\star} \coloneqq \lambda \kappa i.\bigcup_{\langle x,j\rangle \in m\,i} \kappa\, x\, j$$

- Just a combination of the State and Set monads. (In fact, fully determined by something known as the State monad **transformer**, cf. Liang et al. 1995.)

Basic meanings

- Meaning for an indefinite (nondeterministic, but no update):

$$\textbf{A.LING} = \lambda i.\, \left\{ \langle x, i \rangle \mid \textsc{ling}\, x \right\}$$

- And pronouns, where i_0 is the most recently introduced dref in i (deterministic, value returned depends on i, but no update):

$$\textbf{SHE}_0 = \lambda i.\, \left\{ \langle i_0, i \rangle \right\}$$

Introducing drefs

- Introducing drefs can happen modularly:

$$m_{\blacktriangleright} \coloneqq m^{\star}\left(\lambda x i. \{\langle x, i + x\rangle\}\right)$$

- Example with an indefinite:

$$\textbf{A.LING}_{\blacktriangleright} = \lambda i. \left\{\langle x, i + x\rangle \mid \textsc{ling}\, x\right\}$$

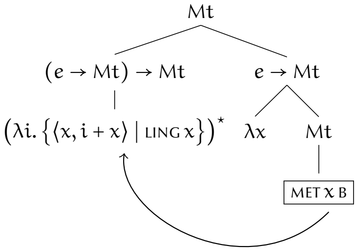
- We can also $\blacktriangleright$-shift simple type e individuals injected into the monad with $\boxed{\cdot}$ (would also work with State monad):

$$\boxed{\textsc{b}}_{\blacktriangleright} = \lambda i. \left\{\langle \textsc{b}, i + \textsc{b}\rangle\right\}$$

- (Possibility of polymorphic drefs for e.g. VP ellipsis.)

Example

- How this works for *Bob met a linguist*▸:



- Gives the expected set of propositions, about different linguists, each tagged with an update:

$$\lambda i.\, \{\langle \textsc{met}\, x\, \textsc{b}, i + x\rangle \mid \textsc{ling}\, x\}$$

- Like the Reader monad's *Bob met Polly*, with nondeterminism. Like the Set monad's *Bob met a linguist*, with index modification.
- Again, *exactly* the same pattern as before.

Getting monadic closure

- Dynamic closure operators have monadic dynamic analogs.
- Negation, type $Mt \to Mt$:

$$\textbf{NOT} = \lambda m i.\, \left\{ \langle \neg \exists \pi \in m\, i : \pi_l, i \rangle \right\}$$

- Universals, type $(e \to Mt) \to Mt$:

$$\textbf{EVERY.BOY} = \lambda \kappa i.\, \left\{ \langle \forall x \in \text{BOY} : \exists \pi \in \kappa\, x\, i : \pi_l, i \rangle \right\}$$

- The results at any κ are deterministic, and encode no update. I.e., they lack side effects – or, in other words, are **pure**.

Where we are

Dynamic semantics

Monads

Monadic dynamic semantics

Features of the monadic account

Modularity

The shape of the grammar and the lexicon

- In standard dynamics, updates are only associated with sentences. In the present account, **any** constituent may encode an update.
- But **needn't**: the dynamic bits of the grammar can be dynamic, but the static parts can stay static. No need to lift the whole thing.
- Ergo, the monadic perspective on dynamics can afford to be more conservative about lexical semantics than standard approaches.

Derived exceptional scope

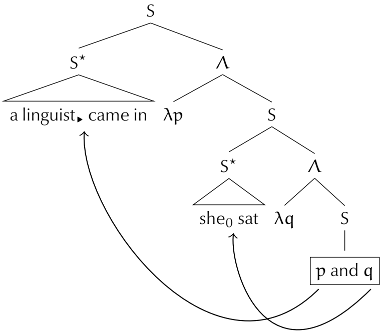
- Every monad's $\star$ is an "associative" operation:

$$(\mathfrak{m}^\star\,(\lambda x.\,\kappa\,x))^\star\,\gamma = \mathfrak{m}^\star\,(\lambda x.\,(\kappa\,x)^\star\,\gamma)$$

- This means **exceptional scope behavior** is a *theorem* of any semantics that uses monads to facilitate composition:
  - Suppose $\mathfrak{m}^\star(\lambda x.\,\kappa\,x)$ is the meaning of some **island**.
  - Associativity means that, even so, $\mathfrak{m}$ can acquire a kind of semantic "scope" over γ's outside the island.

Exceptional scope #1: Dynamic binding

- Remarkably, dynamic binding arises via a kind of **'LF' pied-piping** (cf. Nishigauchi 1990):



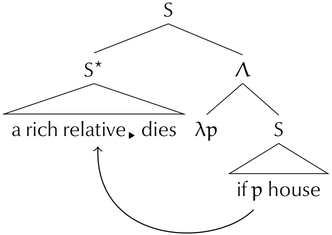
- Result: $\lambda i.\, \left\{ \langle \textsc{came}\, x \wedge \textsc{sat}\, x, i + x \rangle \mid \textsc{ling}\, x \right\}$
- Unlike standard dynamic approaches, this derivation doesn't require a notion of dynamic conjunction.
  - In keeping with the approach I've been advocating, conjunction is boring and interacts with fancy things via $\boxed{\cdot}$ and $\star$.

Exceptional scope #2: Indefinites

- Exceptionally scoping indefinites (e.g., Reinhart 1997):

(4) If [a rich relative of mine dies], I'll inherit a house. $(\exists > \text{if})$

- Exceptional scope is derived, again, by 'LF' pied-piping:



- By associativity, this will end up equivalent to:

$$\textbf{A.RELATIVE}_{\blacktriangleright}^{\star}\ (\lambda x.\ \textsc{if} \ldots) = \lambda i.\left\{\left\langle \textsc{dies}\, x \Rightarrow \textsc{house}, i + x \right\rangle \mid \textsc{relative}\, x\right\}$$

Exceptionally scoping indefinites (cont.)

- Upshot: **unified** take on dynamic binding, exceptional scope. Eludes static, dynamic approaches to indefiniteness.
- Also gives better empirical coverage of exceptionally scoping indefinites than extant accounts (e.g., choice functions).
- E.g., for us exceptional scope really requires scope (i.e., of the island)! So we don't wrongly predict wide-scope-indefinite readings for things like the following (Schwarz 2001):

(5) No candidate$_i$ submitted a paper he$_i$ wrote. (*a > no)

Exceptional scope #3: Proper names

- Proper names can bind pronouns, no matter how embedded:

(6) If e.o. [who hates Walt_i] comes, I'll feel bad for him_i
If e.o. [who hates PETE_j] comes, I won't (feel bad for him_j).

- Predicted by our theory: by associativity, so long as the [island] can scope over the pronoun, the proper name can bind the pronoun.

Exceptional scope #4: Maximal drefs

- **Maximal drefs** contributed by deeply embedded quantifiers:

(7) Everyone heard the rumor that [at most six [senators]$_i$ [supported Cruz's filibuster]$_j$]. It turned out to be erroneous: they$_{i \cap j}$ numbered at least ten.

- Suggests even quantifiers take a kind of exceptional scope.
- Predicted if quantifiers introduce maximal drefs, as is standard in modern dynamic semantics (Kamp & Reyle 1993):

$$\textbf{AT.MOST.SIX.SENATORS} = \lambda \kappa i.\, \{\langle |\textsc{sen} \cap X| \leqslant 6, i + X \rangle\}$$
$$\text{where } X = \textsc{sen} \cap \{x \mid \exists \pi \in \kappa\, x\, i.\, \pi_1\}$$

Where we are

Dynamic semantics

Monads

Monadic dynamic semantics

Features of the monadic account

Modularity

Extension #1: Focus

- Focus usually handled with bidimensional meanings:

$$[\![A\ B]\!]^{o} = [\![A]\!]^{o}\,[\![B]\!]^{o} \qquad [\![A\ B]\!]^{f} = \left\{ f\,x \mid f \in [\![A]\!]^{f}, x \in [\![B]\!]^{f} \right\}$$

- Monadic version (Shan's 2002 pointed powerset monad):

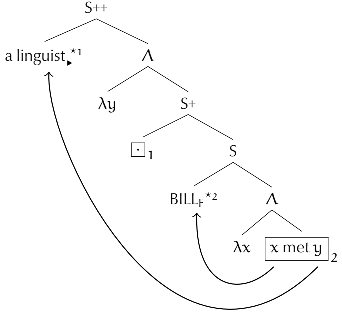
$$\boxed{x} \coloneqq \left\langle x, \{x\} \right\rangle \qquad \langle x, S\rangle^{\star} \coloneqq \lambda\kappa.\left\langle (\kappa\,x)_{l}, \bigcup_{y \in S} (\kappa\,y)_{r} \right\rangle$$

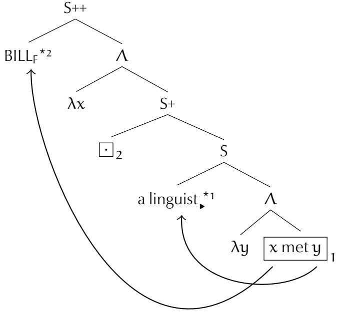
- Meanings for F-marked nodes:

$$x_{F} \coloneqq \left\langle x, \textsc{alt}_{x} \right\rangle$$

Focus (cont.)

- There's nothing else to do! Instead of 2 combinators running around, we'll have 4. But they play nicely together (Charlow 2014).





$M_1 M_2\, t :: i \to \left\{\left\langle\langle t, \{t\}\rangle, i\right\rangle\right\}$ $\qquad$ $M_2 M_1\, t :: \left\langle i \to \{\langle t, i\rangle\}, \left\{i \to \{\langle t, i\rangle\}\right\}\right\rangle$

- This technique is known as **composing applicative functors** (McBride & Paterson 2008). It works for *any* number of monads.

Extension #2: Conventional Implicature

- Negation appears not to interact with nonrestrictive relatives:

(8) I didn't read *Great Expectations*, which is a stone cold classic.

- Potts 2005 proposes a non-compositional two-dimensional semantics to derive this.
- Giorgolo & Asudeh 2012 suggest the **Writer** monad:

$$\boxed{x} \coloneqq x \bullet \top \qquad (x \bullet p)^{\star} \coloneqq \lambda\kappa.\, v \bullet (p \wedge q) \text{ where } v \bullet q = \kappa\, x$$

Conventional implicature (cont.)

- Also comes with a transformer, can be used to roll a big monad that does dynamic binding and 2nd dimensional stuff (and focus!):

$$\mathsf{M}a ::= i \to \{\langle a \bullet t, i \rangle\}$$

- The $\boxed{\cdot}$ operation:

$$\boxed{x} \coloneqq \lambda i. \{\langle x \bullet \top, i \rangle\}$$

- And the $\star$ operation:

$$m^\star \coloneqq \lambda \kappa. \bigcup_{\langle x \bullet p, j \rangle \in m\, i} \{\langle v \bullet (p \wedge q), k \rangle \mid \langle v \bullet q, k \rangle \in \kappa\, x\, j\}$$

- A number of nice results. Feel free to ask about them.

Alternative semantics

- Reader + Set monad, for index-dependence and nondeterminism:

$$\boxed{x} = \lambda i.\left\{\langle x, i\rangle\right\} \qquad\qquad m^\star = \lambda\kappa.\lambda i.\bigcup_{\langle x, j\rangle \in m\, i} \kappa\, x\, i$$

- Still gets exceptional scope. Only the dynamic monad gets dynamic anaphora.
- (It turns out that there's no need to define a combined Reader + Set monad. Simply turning the Reader and Set monads loose is enough, as with Focus.)

Applicatives? Transformers? Functors?

- Monadic $\star$:

$$Ma \to \underbrace{(a \to Mb) \to Mb}_{\text{scope}}$$

- Can always be composed into an applicative functor (sometimes also a monad):

$$M_1 M_2 a \qquad M_2 M_1 a$$

- Functor `fmap` type:

$$(a \to b) \to Fa \to Fb$$

- Flipped:

$$Fa \to \underbrace{(a \to b) \to Fb}_{\text{scope}}$$

Wrapping up

- Go monadic: a shift in perspective (thinking of dynamic semantics in terms of side effects) buys a lot.
- There's empirical and methodological juice:
  - Better coverage (exceptional scope).
  - More extensible, via transformers, applicatives, functors.
- You needn't even go dynamic to reap the fruits. There's something for dyed-in-the-wool static alternative-semanticists, too.

THANKS!

References

Charlow, Simon. 2014. *On the semantics of exceptional scope*: New York University Ph.D. thesis.

Charlow, Simon. 2015. The scope of alternatives. Talk presented at SALT 25.

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.

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.

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

Hamblin, C. L. 1973. Questions in Montague English. *Foundations of Language* 10(1). 41–53.

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

Kamp, Hans & Uwe Reyle. 1993. *From Discourse to Logic*. Dordrecht: Kluwer Academic Publishers.

Kratzer, Angelika & Junko Shimoyama. 2002. Indeterminate pronouns: The view from Japanese. In Yukio Otsu (ed.), *Proceedings of the Third Tokyo Conference on Psycholinguistics*, 1–25. Tokyo: Hituzi Syobo.

Liang, Sheng, Paul Hudak & Mark Jones. 1995. Monad transformers and modular interpreters. In *22nd ACM Symposium on Principles of Programming Languages (POPL '95)*, 333–343. ACM Press.

McBride, Conor & Ross Paterson. 2008. Applicative programming with effects. *Journal of Functional Programming* 18(1). 1–13.

References (cont.)

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.

Nishigauchi, Taisuke. 1990. *Quantification in the theory of grammar*. Dordrecht: Kluwer Academic Publishers.

Partee, Barbara H. 1986. Noun phrase interpretation and type-shifting principles. In Jeroen Groenendijk, Dick de Jongh & Martin Stokhof (eds.), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, 115–143. Dordrecht: Foris.

Potts, Christopher. 2005. *The logic of conventional implicatures*. Oxford: Oxford University Press.

Reinhart, Tanya. 1997. Quantifier scope: How labor is divided between QR and choice functions. *Linguistics and Philosophy* 20(4). 335–397.

Schwarz, Bernhard. 2001. Two kinds of long-distance indefinites. In Robert van Rooy & Martin Stokhof (eds.), *Proceedings of the Thirteenth Amsterdam Colloquium*, 192–197. University of Amsterdam.

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

Shan, Chung-chieh. 2005. *Linguistic side effects*: Harvard University Ph.D. thesis.

Shan, Chung-chieh & Chris Barker. 2006. Explaining crossover and superiority as left-to-right evaluation. *Linguistics and Philosophy* 29(1). 91–134.

References (cont.)

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.

Wadler, Philip. 1992. Comprehending monads. In *Mathematical Structures in Computer Science*, vol. 2 (special issue of selected papers from 6th Conference on Lisp and Functional Programming), 461–493.

Wadler, Philip. 1994. Monads and composable continuations. *Lisp and Symbolic Computation* 7(1). 39–56.

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.