1 Outline

- Continuized CCGs offer a grammar-wide generalization of scope-taking ("ubiquitous scopal pied piping") that brings quantifiers and scope-taker more generally into the compositional fold. Three combinators, **lift**, **triv**, and **scope**, form the backbone of the grammar, allowing scope-takers to interact with their linguistic context. In addition, a **bind** shifter is introduced to facilitate quantificational binding. de Groote 2001

- Extant CCG treatments of E-type anaphora (i.e. cross-sentential and donkey anaphora): Barker & Shan 2008. Motivation for pursuing another approach. BS approach has not-so-good empirical coverage (Charlow 2010). de Groote 2006 does not handle scope-taking or interactions between scope-takers (can be combined with BS regime, but requires two notions of continuations—a scope and a right context—and does not extend to exceptional scope).

- Proposal: replace **lift** and **triv** with options that countenance side effects (Shan 2005). Any side effects regime can be grafted onto a continuized CCG, by replacing **lift** and **triv** with monadic functors (Moggi 1989, Wadler 1992, 1994, 1995, Shan 2002).

- Dynamic semantics is (Shan 2001):[1]
  - State: ability to manipulate the discourse context, i.e. create discourse referents.
  - Nondeterminism: analogizes indefinites to referential expressions. Treats indefinites as referring expressions, though ones which refer indeterminately.

- We provide a general technique for integrating a monadic approach to side effects with continuations-based approaches to scope in CCG. We relate our approach to the ContT monad transformer (Liang et al. 1995). Offers a type-theoretic way to track effects, integrate them into a well-developed CCG framework for scope-taking.

- Therefore, these results are of interest both for the categorial grammarian interested in donkey anaphora and scope-taking, as well as more generally. Any side-effects regime a semanticist thinks is motivated can be accommodated along these lines. Further, because of the inherent modularity, adding side effects necessitates neither fiddling with the basic compositional machinery, nor messing with lexical items which don't exploit side effects.

- Corollary: there is no need to settle on a single ("the") grammar. Different and quite varied side effects regimes can be modularly grafted onto a simple applicative ("pure") core. Lexical entries that would seem incongruous in a flat-footed standard perspective integrate seamlessly in a single grammar.

- Monads as a natural way to extend a continuations-based grammar with tools for dynamic binding and exceptional scope. In the end: you have functional application, plus the functors from whichever monads are implicated in a given language.

- The standard continuations-based perspective of Barker 2002, Shan & Barker 2006, Barker & Shan 2014 is an instantiation of a more general perspective.

- Standard dynamic techniques (DPL, DMG) not reducible to monads.

- Broader question: how this relates to the idea that continuations can simulate any monad (Filinski 1994). I don't understand this result well enough to say anything (Dylan?).

2 Adding side effects

- Standard continuized grammar is, simplifying somewhat, three combinators: **lift**, **triv**, and **scope**. Figure 1.

- Continuized type constructor. Agnostic about directionality. Combined with direction-sensitive mode of combination. See below. $\mathsf{K}ar$ is a meaning which functions as something of type a in a context of type r (the 'result type'). For example, extensional generalized quantifiers have type $\mathsf{K}et$.

$$\mathsf{K}ar ::= (a \to r) \to r$$

- Type-theoretic details here. Dylan: I am not entirely sure the type system makes sense. What I'm after: something basically along the lines of Shan & Barker 2006, where combinators apply to combinators. Interesting property of that system: they use Lift to allow them only one Scope combinator (i.e. the thing on the left can always be the functor). Central question: the proper way to relate the continuations mode slashes with the direct mode slashes. The way I did it in my diss appendix was essentially to have a unimodal grammar, but a more elegant solution would be welcome (and important since this is after all a categorial grammar conference!).

- Adding side effects (Wadler 1994, 1995, Shan 2002): monads. A monad is a triple $\langle \mathsf{M}, \eta, \star \rangle$ of a type constructor M, an injection function η of type $a \to \mathsf{M}a$ (given any type a), and a recipe for sequencing $\star$ of type $\mathsf{M}a \to (a \to \mathsf{M}b) \to \mathsf{M}b$ (given any types a,b).

- Monad laws / punting

Definition 1.

[1] NB: does not characterize all varieties of dynamic semantics. Dynamic treatments following Groenendijk & Stokhof 1990 (e.g. Zimmermann 1991, Dekker 1993, Szabolcsi 2003, de Groote 2006) provide a way for indefinites to extend their binding domain but do not treat indefinites as nondeterministic analogs of proper names.

$$\frac{\Gamma \vdash f : b/a \quad \Delta \vdash e : a}{\Gamma\cdot\Delta \vdash f\,e : b}\,/ \qquad \frac{\Delta \vdash e : a \quad \Gamma \vdash f : a\backslash b}{\Delta\cdot\Gamma \vdash f\,e : b}\,\backslash \qquad \frac{\Delta \vdash m : \mathsf{K}(b/a)r \quad \Gamma \vdash n : \mathsf{K}ar}{\Delta\cdot\Gamma \vdash \mathbf{S}mn : \mathsf{K}br}\,/\!\!/$$

$$\frac{}{\varepsilon \vdash \lambda x.x : a \to a}\,\textsf{triv} \qquad \frac{}{\varepsilon \vdash \lambda xk.k\,x : a \to \mathsf{K}ar}\,\textsf{lift}$$

Figure 1: Continuized CCG without side effects, fixing a result type r.

$$\frac{\Gamma \vdash f : b/a \quad \Delta \vdash e : a}{\Gamma\cdot\Delta \vdash f\,e : b}\,/ \qquad \frac{\Delta \vdash e : a \quad \Gamma \vdash f : a\backslash b}{\Delta\cdot\Gamma \vdash f\,e : b}\,\backslash \qquad \frac{\Delta \vdash m : \mathsf{K}(b/a)r \quad \Gamma \vdash n : \mathsf{K}ar}{\Delta\cdot\Gamma \vdash \mathbf{S}mn : \mathsf{K}br}\,/\!\!/$$

$$\frac{}{\varepsilon \vdash \eta : a \to \mathsf{M}a}\,\textsf{triv} \qquad \frac{}{\varepsilon \vdash (\star) : \mathsf{M}a \to \mathsf{K}a\mathsf{M}r}\,\textsf{lift}$$

Figure 2: Continuized CCG with side effects, fixing a monad $\langle \mathsf{M}, \eta, \star \rangle$ and a result type r.

- The key to connecting monads with continuations is realizing that the type of $(\star)$ can be rewritten using the continuized type constructor as $\mathsf{M}a \to \mathsf{K}a\mathsf{M}b$

- Relating monads to continuized grammars: identify **lift** with $\star$, **triv** with η. But **scope** stays the same.

- Regular lifting is a theorem, though the types are further specified:

  Fact 1.

3 Finding the dynamic monad

- The meat of PLA (Dekker 1994), translated into a compositional framework: sentences denote relations on sequences. Non-empty relations correspond to truth. Non-functional pairs in the relation correspond to nondeterminism introduced by indefinites (and perhaps disjunction). Conjunction corresponds to relation composition, which pipes the sequences output by the left conjunct to the right conjunct.

$$\begin{aligned} [\![\text{a linguist}]\!] &= \lambda ki.\bigcup_{x \in \mathsf{ling}} k\,x\,(i+x) \\ [\![\text{a linguist left}]\!] &= \lambda i.\{i+x : x \in \mathsf{ling} \wedge x \in \mathsf{left}\} \end{aligned}$$

- Two key bits: state modification for introducing drefs, nondeterminism to allow for failure and referring treatment of indefinites. A different perspective on this: treating nondeterminism and state modification as side effects, within a functional programming setting for side effects.

- Monad for state (generalization of monad for environment-sensitivity). Assume that γ is the type of "contexts of evaluation". For our purposes, we might think of γ as inhabited by *sequences of discourse referents*.

  Definition 2 (The State monad).

$$\begin{aligned} \mathsf{M}a &::= \gamma \to a \times \gamma \\ \eta\,x &:= \lambda i.\langle x, i\rangle \\ m \star k &:= \lambda i.k\,(mi)_0\,(mi)_1 \end{aligned}$$

- Given our identification of γ with the set of sequences of discourse referents, a natural operation to suppose as associated with dref introduction is sequence extension (cf. de Groote 2006, Unger 2012, Charlow 2014). These definitions rely on the notion of extending a sequence (e.g., if $i := abcd$, $i + e = abcde$) and retrieving the last, i.e. most topical, discourse reference (e.g., if $i := abcde$, $i_\top = e$).[2]

  Definition 3 (Dref introduction).

$$m^{\rhd} := m \star \lambda xi.\langle x, i+x\rangle$$

  Definition 4 (Dref retrieval).

$$\textsf{\textbf{he}} := \lambda i.\langle i_\top, i\rangle$$

- An example, *Al left* (call this **X**):

$$(\eta\,\mathsf{a})^{\rhd} \star \lambda x.\eta\,(\mathsf{left}\,x) = \lambda i.\langle \mathsf{left}\,\mathsf{a}, i+\mathsf{a}\rangle$$

- Pronoun sentence *he was tired* (call this **Y**):

$$\textsf{\textbf{he}} \star \lambda x.\eta\,(\mathsf{tired}\,x) = \lambda i.\langle \mathsf{tired}\,i_\top, i\rangle$$

- Sequencing the two. The dref introduced by the proper name in the first sentence is accessed by the pronoun in the second.

$$\begin{aligned} &\mathbf{X} \star \lambda p.\mathbf{Y} \star \lambda q.\eta\,(p \wedge q) \\ &\quad = \lambda i.\langle \mathsf{left}\,\mathsf{a} \wedge \mathsf{tired}\,\mathsf{a}, i+\mathsf{a}\rangle \end{aligned}$$

- So we have state modification as a side effect. To say something about indefinites, i.e. to allow them to refer and introduce drefs nondeterministically, we need to enrich the state monad with nondeterministic side effects. The monad for nondeterminism is the Set monad, given in Definition 5:

  Definition 5 (The Set monad).

$$\begin{aligned} \mathsf{M}a &::= a \to t \\ \eta\,x &:= \{x\} \\ m \star k &:= \bigcup_{x \in m} k\,x \end{aligned}$$

[2] This is an extremely crude measure of topicality, but it will suffice to illustrate the main points.

- Use StateT to stitch the two together. Given any monad $\mathcal{M} = \langle \mathsf{L}, \eta_\mathsf{L}, \star_\mathsf{L}\rangle$, StateT is a recipe for building a new monad with which adds State-type functionality to $\mathcal{M}$:[3]

  Definition 6 (The StateT monad transformer).

  $$\begin{aligned} \mathsf{M}\,a &::= \gamma \to \mathsf{L}\,(a \times \gamma) \\ \eta\,x &:= \lambda i.\,\eta_\mathsf{L}\,\langle x, i\rangle \\ m \star k &:= \lambda i.\, m\, i \star_\mathsf{L} \lambda \pi.\, k\, \pi_0\, \pi_1 \end{aligned}$$

  Definition 7 (The State_Set monad).

  $$\begin{aligned} \mathsf{M}\,a &::= \gamma \to (a \times \gamma) \to t \\ \eta\,x &:= \lambda i.\,\{\langle x, i\rangle\} \\ m \star k &:= \lambda i.\, \bigcup_{\pi \in m i} k\, \pi_0\, \pi_1 \end{aligned}$$

- Static lexicon, dynamic lexicon

- Modular treatment of binding.

  $$\begin{aligned} \text{Previous}: &\quad \mathbf{bind}\, m := \lambda k.\, m\,(\lambda x.\, k\, x\, x) \\ \text{Proposal}: &\quad \mathbf{bind}\, m := \lambda k.\, m\,(\lambda x i.\, k\, x\,(i + x)) \end{aligned}$$

- Summing up: three combinators for "order-insensitive" (i.e. continuized combination). **unit**, **run**, **bind**. Compare Figures 1 and 2.

4 Examples

- Some upshots: no dynamic conjunction, completely standard model theory (cf. de Groote 2006). "Contexts of evaluation" are constructed on the fly.

- Cross-sentential anaphora. Let us assume that *scope islands*, e.g., tensed clauses, need to be evaluated—i.e., lowered (cf. Barker 2002, Barker & Shan 2008). In practice, this means a tensed clause must pass through a stage where it denotes something of type $\mathsf{M}\,t$. There are a variety of options for enforcing this syntactically, but here we concentrate on the semantic upshots of forced evaluation.[4] the indefinite's side effects influence the evaluation of the second clause, even as the indefinite scopes within its clause.

  $$\begin{aligned} &\mathbf{a.man.left} \star \lambda p.\, \mathbf{he.tired} \star \lambda q.\, \eta\,(p \wedge q) \\ &\quad = \mathbf{a.man} \star \lambda x.\, \eta\,(\mathsf{left}\, x \wedge \mathsf{tired}\, x) \end{aligned}$$

- Compare universals. After ending the derivation at the clause boundary, we're left with a pure computation. The universal's side effects have died on evaluation.

  $$\eta\,(\forall x.\, \mathsf{ling}\, x \Rightarrow \mathsf{left}\, x)$$

- Donkey anaphora works similarly. Take the following. The restrictor c here acquires a kind of monadic scope, via $\star$, over the nuclear scope k. This means any side effects inside c influence the context of evaluation for k. However, once k is grabbed, the wide-scoping negation discharges side effects (as is standard in dynamic systems).

  $$[\![\text{every}]\!] := \lambda c k.\, \mathbf{not}\,(\mathbf{a}\, c \star \lambda x.\, \mathbf{not}\,(k\, x))$$

- Islands: a clause must denote a $\mathsf{M}\,t$

5 Discussion

-

- Compare PLA, where only sentences are imbued with context change potential.

- Variable-free, directly compositional (Jacobson 1999).

- Effects recognized in the types.

- Theory extends to scope islands, wide range of exceptional binding configurations Charlow 2014.

- Extends to pair-list phenomena, functional quantification: Bumford to appear

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. 2008. Donkey Anaphora is In-Scope Binding. *Semantics & Pragmatics* 1(1). 1–46. http://dx.doi.org/10.3765/sp.1.1.

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and Natural Language*. Oxford: Oxford University Press.

Bumford, Dylan. to appear. Incremental quantication and the dynamics of pair-list phenomena. *Semantics & Pragmatics* 8(9).

Charlow, Simon. 2010. Two kinds of binding out of DP. Unpublished ms.

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

Dekker, Paul. 1993. *Transsentential meditations: ups and downs in dynamic semantics*: University of Amsterdam Ph.D. thesis.

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.

Filinski, Andrzej. 1994. Representing Monads. In *Proceedings of the 21st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 446–457. New York: ACM Press.

Groenendijk, Jeroen & Martin Stokhof. 1990. Dynamic Montague Grammar. In Laszlo Kalman & Laszlo Polos (eds.), *Proceedings of the Second Symposium on Logic and Language*, 3–48. Budapest: Eötvös Loránd University Press.

[3] Fn. about SetT

[4] In terms of LF, forcing evaluation of a scope island corresponds to disallowing QR out of the scope island.

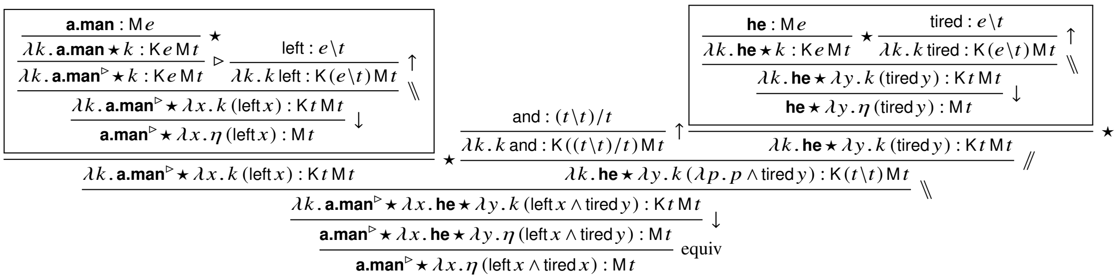
a.man : Me
λk . a.man ⋆ k : KeMt
λk . a.man^▷ ⋆ k : KeMt
left : e\t
λk . k left : K(e\t)Mt
λk . a.man^▷ ⋆ λx . k (left x) : KtMt
a.man^▷ ⋆ λx . η (left x) : Mt
λk . a.man^▷ ⋆ λx . k (left x) : KtMt
and : (t\t)/t
λk . k and : K((t\t)/t)Mt
he : Me
λk . he ⋆ k : KeMt
tired : e\t
λk . k tired : K(e\t)Mt
λk . he ⋆ λy . k (tired y) : KtMt
he ⋆ λy . η (tired y) : Mt
λk . he ⋆ λy . k (tired y) : KtMt
λk . he ⋆ λy . k (λp . p ∧ tired y) : K(t\t)Mt
λk . a.man^▷ ⋆ λx . he ⋆ λy . k (left x ∧ tired y) : KtMt
a.man^▷ ⋆ λx . he ⋆ λy . η (left x ∧ tired y) : Mt
a.man^▷ ⋆ λx . η (left x ∧ tired x) : Mt
⋆
▷
↑
⫽
↓
equiv

Figure 3: Cross-sentential anaphora.

de Groote, Philippe. 2001. Type raising, continuations, and classical logic. In Robert van Rooy & Martin Stokhof (eds.), *Proceedings of the Thirteenth Amsterdam Colloquium*, 97–101. University of Amsterdam.

de Groote, Philippe. 2006. Towards a Montagovian account of dynamics. In Masayuki Gibson & Jonathan Howell (eds.), *Proceedings of Semantics and Linguistic Theory 16*, 1–16. Ithaca, NY: Cornell University.

Jacobson, Pauline. 1999. Towards a Variable-Free Semantics. *Linguistics and Philosophy* 22. 117–184. http://dx.doi.org/10.1023/A:1005464228727.

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.

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.

Shan, Chung-chieh. 2001. A Variable-Free Dynamic Semantics. In Robert van Rooy & Martin Stokhof (eds.), *Proceedings of the Thirteenth Amsterdam Colloquium*, 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. http://dx.doi.org/10.1007/s10988-005-6580-7.

Szabolcsi, Anna. 2003. Binding on the Fly: Cross-Sentential Anaphora in Variable-Free Semantics. In Geert-Jan M. Kruijff & Richard T. Oehrle (eds.), *Resource-Sensitivity, Binding and Anaphora*, 215–227. Dordrecht: Kluwer Academic Publishers.

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. 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. 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.

Zimmermann, Thomas Ede. 1991. Dynamic logic and case quantification. In Martin Stokhof, Jeroen Groenendijk & David Beaver (eds.), *Quantification and Anaphora I* (DYANA Deliverable R2.2.A), 191–195.