NASSLLI Workshop
New type-theoretic tools in natural language semantics
Thursday 28 June 2018, 3PM CMU

Scope in Natural Language: Why Monads aren't Enough

Chris Barker
New York University, Department of Linguistics

Version of June 28, 2018.

Plan

- Dependent types, applicatives, monads, what are we doing?
- Scope in Natural Language (new: algebraic presentation)
  - type operators
  - decidability
- Empricial challenges
  - WH question formation, relative clause formation
  - Recursive scope: *some of the same*, Andrews Amalgams

What are we doing?

- What algebraic structure best characterizes which natural language phenomena?
- Is the logic of presupposition intuitionistic or classical?
- Does intenstionality call for a monad or a comonad?
- Do we need monads, or are applicatives what we really care about?

Type operator	Plain	Fancy	Application
Writer monad	A	$A \times B$	supplementals
Reader monad	A	$B \to A$	simple binding
State monad	A	$B \to (A \times B)$	binding
Continuation monad	A	$(A \to B) \to B$	simple scope
Continuations	A	$(A \to B) \to C$	scope

Draw circles

Scope

(1) Ann saw Bill.

(2) Ann saw everyone. **everyone**$(\lambda x.$**saw ann** $x)$

(3) Someone saw everyone.

(4) Ann saw *who*?

(5) Who did Ann see __?

(6) That’s the book [the author of which] I met last night.

(7) Ann ate something, but I don’t know what she ate.

(8) Ann ate something, but I don’t know what __. sluice

(9) Ann ate [I don’t know what __] yesterday.

Where we're headed on the Barendregt cube

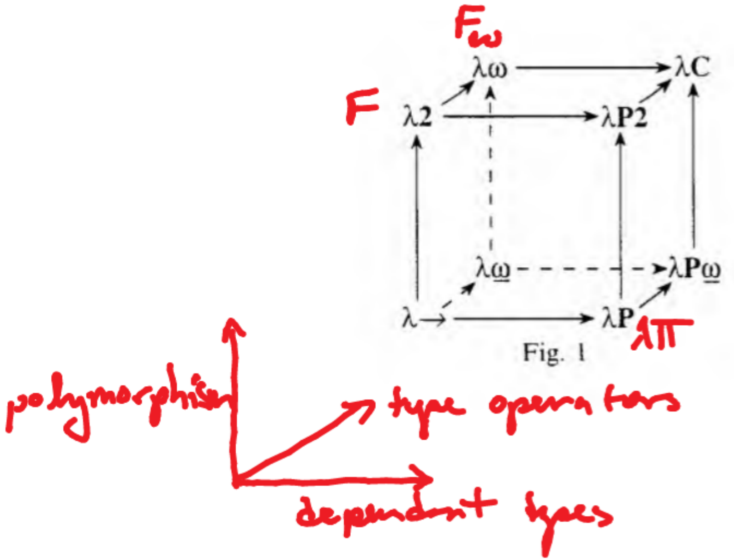


Fig. 1

- Barendregt 1991 *J. of Functional Programming* **1.2**:125–154
- Polymorphism: terms depend on types:

$$\big((\Lambda A \lambda x{:}A.x){:}(\forall A.A \to A)\big)[\mathtt{Int}] = \lambda x{:}\mathtt{Int}.x$$

- Dependent types: types depend on terms
- Type operators: types depend on types: $\mathrm{C}_{\mathrm{t}} e = (e \to \mathrm{t}) \to \mathrm{t}$

Lambek's type logic NL: the logic of external merge

Substructural: without Exchange, '$\supset$' splits into '$\backslash$' and '$/$':

- **Atomic formulas**: $\mathcal{A}\mathrm{t} = \mathsf{DP} \,|\, \mathsf{S} \,|\, \mathsf{N} \,|\, \mathsf{Q}$
- **Formulas:** $\mathcal{F} = \mathcal{A}\mathrm{t} \,|\, \mathcal{F}\backslash\mathcal{F} \,|\, \mathcal{F}/\mathcal{F} \,|\, \mathcal{F} \bullet \mathcal{F}$
- **Sequents:** $\mathcal{F} \vdash \mathcal{F}$
- **Axiom schema:** $\mathrm{A} \vdash \mathrm{A}$
- **Logical rules:**

$$\text{(residuation)} \qquad \mathrm{B} \vdash \mathrm{A}\backslash\mathrm{C} \quad \textsf{iff} \quad \mathrm{A} \bullet \mathrm{B} \vdash \mathrm{C} \quad \textsf{iff} \quad \mathrm{A} \vdash \mathrm{C}/\mathrm{B}$$

$$\text{(transitivity)} \qquad \frac{\mathrm{A} \vdash \mathrm{B} \qquad \mathrm{B} \vdash \mathrm{C}}{\mathrm{A} \vdash \mathrm{C}}\,\textsf{CUT}$$

"$\mathrm{A} \vdash \mathrm{B}$" means
"any expression of type A is also an expression of type B"

Unpacking residuation

(residuation) $\quad B \vdash A\backslash C \quad \text{iff} \quad A \bullet B \vdash C \quad \text{iff} \quad A \vdash C/B$

$$\frac{B \vdash A\backslash C}{A \bullet B \vdash C} \qquad \frac{A \bullet B \vdash C}{A \vdash C/B}$$

$$\frac{A \bullet B \vdash C}{B \vdash A\backslash C} \qquad \frac{A \vdash C/B}{A \bullet B \vdash C}$$

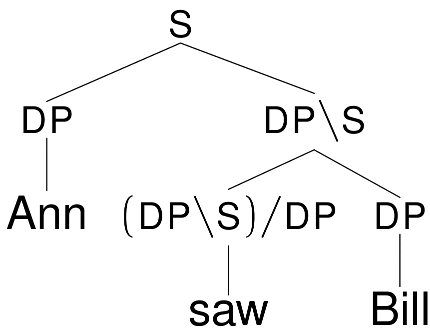
$$\frac{\mathrm{DP} \bullet \textbf{left} \vdash \mathrm{S}}{\textbf{left} \vdash \mathrm{DP}/\mathrm{S}}$$

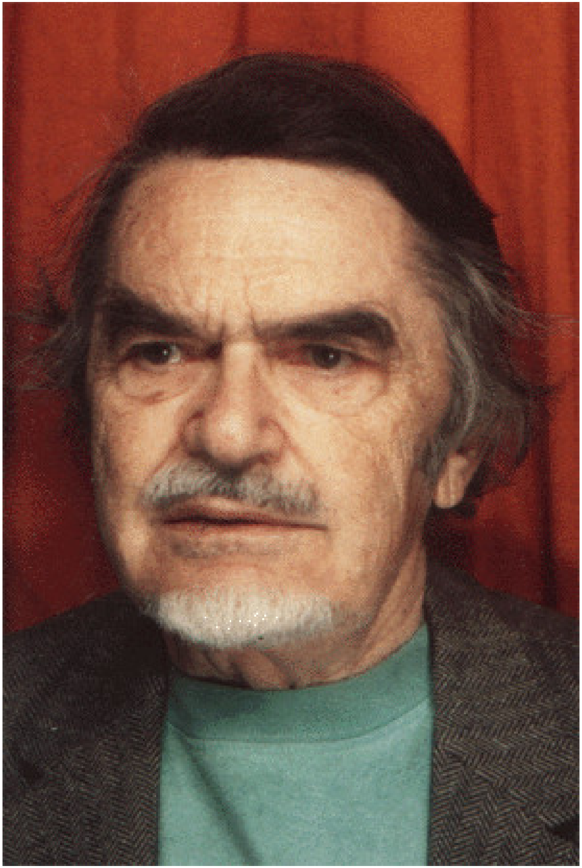
Sample derivation of *Ann saw Bill*

Assume *Ann* and *Bill* have type DP and *saw* has type (DP\S)/DP:

$$\cfrac{\cfrac{\cfrac{\cfrac{}{(\mathrm{DP}\backslash \mathrm{S})/\mathrm{DP} \vdash (\mathrm{DP}\backslash \mathrm{S})/\mathrm{DP}}\ \text{AXIOM}}{(\mathrm{DP}\backslash \mathrm{S})/\mathrm{DP} \bullet \mathrm{DP} \vdash \mathrm{DP}\backslash \mathrm{S}}\ \text{RESIDUATION}}{\mathrm{DP} \bullet ((\mathrm{DP}\backslash \mathrm{S})/\mathrm{DP} \bullet \mathrm{DP}) \vdash \mathrm{S}}\ \text{RESIDUATION}}{\textit{Ann} \bullet (\textit{saw} \bullet \textit{Bill}) \vdash \mathrm{S}}$$

It’s the logic of external merge:



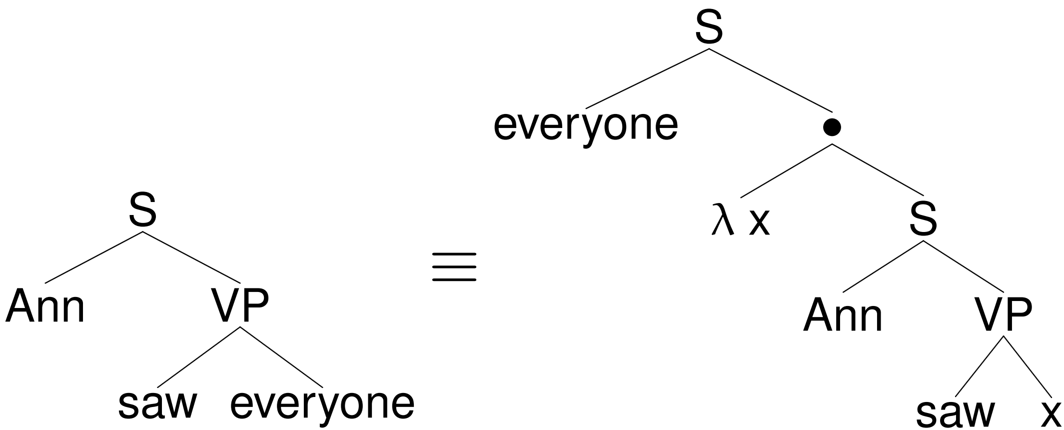


Joachim Lambek

Quantifier Raising as a logical inference

- Montague 1973: Quantifying In: (3065 citations)
- May 1978,1985: Quantifier Raising (QR): (3286 citations)

Montague $\downarrow$ $$\frac{\text{everyone}(\lambda x.\text{Ann saw } x) \vdash S}{\text{Ann saw everyone} \vdash S}$$ $\uparrow$ May



Richard Montague

Robert May

$\mathbf{NL}_{\mathrm{QR}}$, the logic of scope

- **Atomic formulas**: $\mathcal{A}t = \mathsf{DP} \,|\, \mathsf{S} \,|\, \mathsf{N} \,|\, \mathsf{Q}$
- **Variables**: $\mathcal{V} = x \,|\, y \,|\, z \,|\, x' \,|\, x'' \,|\, x''', \ldots$
- **Formulas:** $\mathcal{F} = \mathcal{A}t \,|\, \mathcal{F}\backslash\mathcal{F} \,|\, \mathcal{F}/\mathcal{F} \,|\, \mathcal{F} \bullet \mathcal{F} \,|\, \mathcal{V} \,|\, \lambda\mathcal{V}\mathcal{F}$
- **Sequents:** $\mathcal{F} \vdash \mathcal{F}$
- **Axioms:** $A \vdash A$
- **Logical rules:**

(residuation) $B \vdash A\backslash C$ iff $A \bullet B \vdash C$ iff $A \vdash C/B$

(transitivity)
$$\frac{A \vdash B \qquad B \vdash C}{A \vdash C}\ \mathsf{CUT}$$

(Quantifier Raising) $B[A] \vdash C$ iff $A \bullet \lambda x B[x] \vdash C$

Sample derivation: *Ann saw everyone*

Assume *everyone* has type $\mathsf{S(DP\backslash S)}$, the traditional type of an (extensional) generalized quantifier:

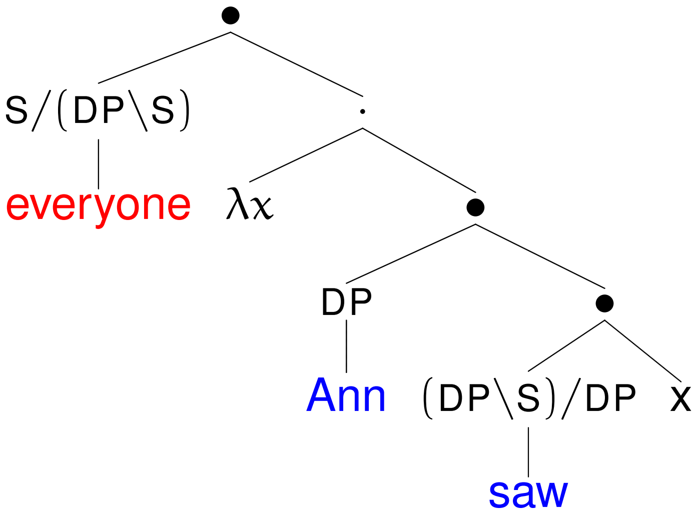
$$\vdots$$

$$\dfrac{\dfrac{\dfrac{\mathsf{DP} \bullet ((\mathsf{DP\backslash S})/\mathsf{DP} \bullet \mathsf{DP}) \vdash \mathsf{S}}{\mathsf{DP} \bullet \lambda x(\mathsf{DP} \bullet ((\mathsf{DP\backslash S})/\mathsf{DP} \bullet x)) \vdash \mathsf{S}}\ \mathrm{QR}}{\lambda x(\mathsf{DP} \bullet ((\mathsf{DP\backslash S})/\mathsf{DP} \bullet x)) \vdash \mathsf{DP\backslash S}}\ \mathrm{RES} \qquad \dfrac{\dfrac{\mathsf{S}/(\mathsf{DP\backslash S}) \vdash \mathsf{S}/(\mathsf{DP\backslash S})}{\mathsf{S}/(\mathsf{DP\backslash S})/(\mathsf{DP\backslash S}) \vdash \mathsf{S}}\ \mathrm{RE}}{\mathsf{DP\backslash S} \vdash (\mathsf{S}/(\mathsf{DP\backslash S})/)\mathsf{S}}\ \mathrm{RE}}{\lambda x(\mathsf{DP} \bullet ((\mathsf{DP\backslash S})/\mathsf{DP} \bullet x)) \vdash (\mathsf{S}/(\mathsf{DP\backslash S})/)\mathsf{S}}\ \mathrm{CU}$$

$$\dfrac{\dfrac{\dfrac{\lambda x(\mathsf{DP} \bullet ((\mathsf{DP\backslash S})/\mathsf{DP} \bullet x)) \vdash (\mathsf{S}/(\mathsf{DP\backslash S})/)\mathsf{S}}{\color{red}{(\mathsf{S}/(\mathsf{DP\backslash S})/) \bullet \lambda x(\mathsf{DP} \bullet ((\mathsf{DP\backslash S})/\mathsf{DP} \bullet x))} \vdash \mathsf{S}}\ \mathrm{RES}}{\mathsf{DP} \bullet ((\mathsf{DP\backslash S})/\mathsf{DP} \bullet \mathsf{S}/(\mathsf{DP\backslash S})/) \vdash \mathsf{S}}\ \mathrm{QR}}{\textit{Ann} \bullet (\textit{saw} \bullet \textit{everyone}) \vdash \mathsf{S}}$$

The structure of the red type is given on the next slide.

A type from the middle of the derivation that tells the story



Arg on left; this is supposed to look highly familiar to linguists;

Type operators

- So $\mathtt{DP}$, $\mathtt{DP}\backslash\mathtt{S}$, $\mathtt{DP} \bullet (\mathtt{DP}\backslash\mathtt{S})$ are types.
- What about $\lambda x(\mathtt{DP} \bullet ((\mathtt{DP}\backslash\mathtt{S})/\mathtt{DP} \bullet x))$?
- An expression (term) of type $\mathtt{DP}\backslash\mathtt{S}$ maps any object of type $\mathtt{DP}$ onto an object of type $\mathtt{S}$.
- So $\mathtt{DP}\backslash\mathtt{S}$ is the type of an object-level function.
- $\lambda x(\mathtt{DP} \bullet ((\mathtt{DP}\backslash\mathtt{S})/\mathtt{DP} \bullet x))$ is a *type operator*.
- It maps any type into a type.
- Systems w/type operators = rear face of the Barendregt cube (systems with dependent types form the right face)
  - λ_ω, the simply-typed lambda calculus with type operators
  - System F_ω, the higher-order polymorphic λ calclus

See Pierce 2002, especially chapters 29 and 30

Some properties of $\mathbf{NL}_{QR}$

- Cut elimination
- Sound and complete wrt the usual relational semantics
- Decidable. This is surprising:
  - $A \vdash B$ iff
    $A \bullet \lambda x x \vdash B$ iff
    $(A \bullet \lambda x x) \bullet \lambda x x \vdash B$ ...
- Proof strategy
  - Easy: QR doesn't interfere with Lambek's proof
  - Soundness and completeness not trivial. Simulate embedding of the λ-calculus in combinatory logic.
  - Decidability, in the equivalent sequent presentation:
    * Each instance of residuation elimiantes one slash.
    * Each instance of QR can be associated with a unique instance of residuation.
    * Finite number of slashes in conclusion sequent.

Barker (under revision); extends to overt syntactic movement

Connection with applicatives and monads?

What do we need to have an applicative?

- Type operation: $\mathcal{C}A \Rightarrow (A \to B) \to B$
- unit ("pure"): $\rho{:}A \to \mathcal{C}A$
- circled star thingie: $\star{:}\mathcal{C}(A \to B) \to (\mathcal{C}A) \to \mathcal{C}B$

Theorems of the logic:

- $A \vdash B/(A\backslash B)$ "Lift"
- $C/((A\backslash B)\backslash C) \vdash (C/(A\backslash C))\backslash(C/(B\backslash C))$

Does it obey the applicative laws?

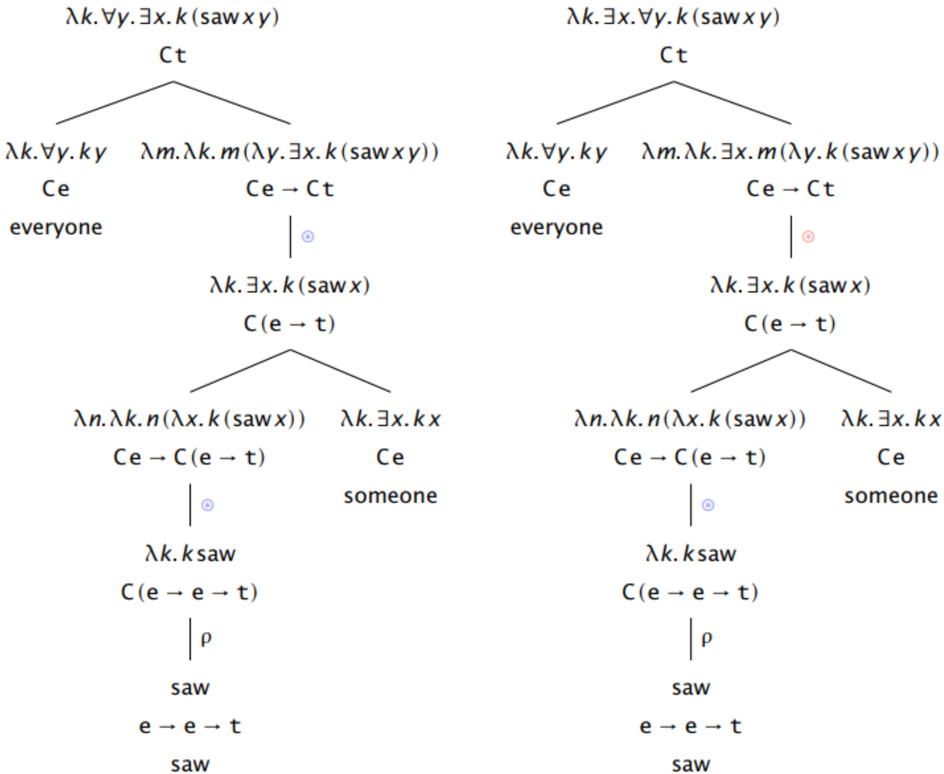
Self-composable, scope ambiguity, finite readings

Example: simple binding by simulating a Reader monad

- *Ann saw Bill* (from above): $\mathrm{DP} \bullet ((\mathrm{DP}\backslash\mathrm{S})/\mathrm{DP} \bullet \mathrm{DP}) \vdash \mathrm{S}$
- Assume the pronoun *him* has type $(\mathrm{DP}\backslash\mathrm{S})/(\mathrm{DP}\backslash\mathrm{S})$.
- *Ann saw him*: $\mathrm{DP} \bullet ((\mathrm{DP}\backslash\mathrm{S})/\mathrm{DP} \bullet (\mathrm{DP}\backslash\mathrm{S})/(\mathrm{DP}\backslash\mathrm{S})) \vdash \mathrm{DP}\backslash\mathrm{S}$
- Curry-Howard proof labeling: $\lambda x.\mathbf{saw}\, x\, \mathbf{ann}$.

Why it's important to have decidability...

Scope interactions, refresher



Charlow and Bumford: “lexical types drive the type shifting”

Andrews Amalgams: ellipsis to a containing continuation

Johnson 2013:

a. Sally will eat something today, but I don't know what __.

b. Sally will eat [I don't know what __] today.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{idk} \bullet (\mathrm{what} \bullet \mathrm{DP}\backslash\!\!\backslash \mathrm{S}) \vdash \mathrm{S}}{\mathrm{DP}\backslash\!\!\backslash \mathrm{S} \circ \lambda x(\mathrm{idk} \bullet (\mathrm{what} \bullet x)) \vdash \mathrm{S}}\,\lambda}{\lambda x(\mathrm{idk} \bullet (\mathrm{what} \bullet x)) \vdash (\mathrm{DP}\backslash\!\!\backslash \mathrm{S})\backslash\!\!\backslash \mathrm{S}}\,\backslash\!\!\backslash R \qquad \mathrm{G} \vdash \mathrm{G}}{\mathrm{G}/\!\!/((\mathrm{DP}\backslash\!\!\backslash \mathrm{S})\backslash\!\!\backslash \mathrm{S}) \circ \lambda x(\mathrm{idk} \bullet (\mathrm{what} \bullet x)) \vdash \mathrm{G}}\,/\!\!/L}{\mathrm{AMALGAM} \circ \lambda x(\mathrm{idk} \bullet (\mathrm{what} \bullet x)) \vdash \mathrm{G}}\,\lambda}{\mathrm{idk} \bullet (\mathrm{what} \bullet \mathrm{AMALGAM}) \vdash \mathrm{G}}\,\lambda$$

$$\dfrac{\dfrac{\dfrac{\lambda y(\mathrm{idk} \bullet (\mathrm{what} \bullet y)) \vdash (\mathrm{DP}\backslash\!\!\backslash \mathrm{S})\backslash\!\!\backslash \mathrm{S} \qquad \mathrm{G} \circ \lambda x(\mathrm{Sally} \bullet (\mathrm{ate} \bullet x)) \vdash \mathrm{S}}{(\mathrm{G}/\!\!/((\mathrm{DP}\backslash\!\!\backslash \mathrm{S})\backslash\!\!\backslash \mathrm{S}) \circ \lambda y(\mathrm{idk} \bullet (\mathrm{what} \bullet y))) \circ \lambda x(\mathrm{Sally} \bullet (\mathrm{ate} \bullet x)) \vdash \mathrm{S}}\,/\!\!/L}{(\mathrm{idk} \bullet (\mathrm{what} \bullet \mathrm{AMALGAM})) \circ \lambda x(\mathrm{Sally} \bullet (\mathrm{ate} \bullet x)) \vdash \mathrm{S}}\,\lambda, \mathrm{LEX}}{\mathrm{Sally} \bullet (\mathrm{ate} \bullet (\mathrm{idk} \bullet (\mathrm{what} \bullet \mathrm{AMALGAM}))) \vdash \mathrm{S}}\,\lambda$$

$\mathrm{G} \equiv \mathrm{S}/\!\!/(\mathrm{DP}\backslash\!\!\backslash \mathrm{S})$ (i.e., scope-taking DP, a generalized quantifier)

The full power of continuations (indexed applicatives)

Assume AMALGAM has type $\mathrm{Q}/(\mathrm{GAP}\backslash\mathrm{S})$.

Sketch of *Sally ate [I don't know what* AMALGAM*]*:

$$\dfrac{\dfrac{\dfrac{\lambda y(\mathrm{idk}\cdot(\mathrm{what}\cdot y)) \vdash (\mathrm{DP}\backslash\!\!\backslash\mathrm{S})\backslash\!\!\backslash\mathrm{S} \qquad \mathrm{G}\circ\lambda x(\mathrm{Sally}\cdot(\mathrm{ate}\cdot x))\vdash\mathrm{S}}{(\mathrm{G}/\!\!/((\mathrm{DP}\backslash\!\!\backslash\mathrm{S})\backslash\!\!\backslash\mathrm{S})\circ\lambda y(\mathrm{idk}\cdot(\mathrm{what}\cdot y)))\circ\lambda x(\mathrm{Sally}\cdot(\mathrm{ate}\cdot x))\vdash\mathrm{S}}\,/\!\!/L}{(\mathrm{idk}\cdot(\mathrm{what}\cdot\mathrm{AMALGAM}))\circ\lambda x(\mathrm{Sally}\cdot(\mathrm{ate}\cdot x))\vdash\mathrm{S}}\,\equiv,\mathrm{LEX}}{\mathrm{Sally}\cdot(\mathrm{ate}\cdot(\mathrm{idk}\cdot(\mathrm{what}\cdot\mathrm{AMALGAM})))\vdash\mathrm{S}}\,\equiv$$

- Type constructors lifting A into $(A \to B) \to B$ are not enough.
- Also need A into $(A \to B) \to C$
- Not a monad. Not an applicative.

Details in my 2013 *Linguistics and Philosopy* paper on sluicing

Conclusions

Here's the logic:

$$\text{(external merge)} \qquad B \vdash A\backslash C \quad \textsf{iff} \quad A \bullet B \vdash C \quad \textsf{iff} \quad A \vdash C/B$$

$$\text{(scope)} \qquad B[A] \vdash C \quad \textsf{iff} \quad A \bullet \lambda x B[x] \vdash C$$

- Provides the full power of continuations
- Simulates applicatives
- Full factorial scope ambiguity
- Soundness and completeness, decidability
- The lexical types drive the proof search

THANKS!

(For today's talk, especially thanks to Colin, Simon, and Dylan)

Selected References

Barendregt, Henk. 1991. [title] *J. of Functional Programming* **1.2**:125–154

Barker, Chris. 2007. Parasitic Scope. *L& P* **30.4**: 407–444.

Barker, Chris. 2015. Scope. In Shalom Lappin and Chris Fox (eds). *Handbook of Contemporary Semantics, 2d edition*. Wiley-Blackwell.

Barker, Chris. 2018. [Soundness, completeness, and decidability for NL_λ] Available at semanticsarchive.net; currently under revision.

Barker, Chris and Chung-chieh Shan. 2014. *Continuations and Natural Language*. Oxford.

Johnson, Kyle. 2013. Recoverability of deletion. In Kuniya Nasukawa and Henk C. van Riemsdijk (eds). *Identity Relations in Grammar*. Berlin: Mouton de Gruyter (Studies in Generative Grammar series).

Kiselyov, Oleg, Shan, Chung-chieh. 2014. Continuation hierarchy and quantifier scope. In *Formal Approaches to Semantics and Pragmatics*, Eric McCready, Katsuhiko Yabushita, Kei Yoshimoto (eds).

Lambek, Joachim. 1958. The mathematics of sentence structure. *The American Mathematical Monthly* **60.3**: 154–170.

Morrill, Glyn, Oriol Valent n, and Mario Fadda. 2011. The displacement calculus. Journal of Logic, Language and Information 20(1):148.

Solomon, Mike. 2009. Partitives and the semantics of *same*. se