Generalizing continuations

Semantics II

April 30 & May 3, 2018

Applicatives (McBride & Paterson 2008, Kiselyov 2015, Charlow 2017)

A type constructor F is applicative if it supports ρ and $\circledast$ with these types...

$$\rho : a \to F\,a \qquad\qquad \circledast : F\,(a \to b) \to F\,a \to F\,b$$

...Where ρ is a trivial way to inject something into the richer type characterized by F, and $\circledast$ is function application lifted into F.

Some concrete applicatives

For $\mathsf{G}\,a ::= \mathsf{g} \to a$, we have an Applicative for assignment-dependence:

$$\underbrace{\rho\, x := \lambda g.\, x}_{\text{cf. } [\![\text{John}]\!] := \lambda g.\mathrm{j}} \qquad \underbrace{m \circledast n := \lambda g.\, m g\,(n g)}_{\text{cf. } [\![\alpha\ \beta]\!] := \lambda g.[\![\alpha]\!]\, g\,([\![\beta]\!]\, g)}$$

For $\mathsf{S}\,a ::= a \to \{0, 1\}$, we have an Applicative for alternatives:

$$\underbrace{\rho\, x := \{x\}}_{\text{cf. } [\![\text{John}]\!] := \{\mathrm{j}\}} \qquad \underbrace{m \circledast n := \{f\, x \mid f \in m,\, x \in n\}}_{\text{cf. } [\![\alpha\ \beta]\!] := \{f\, x \mid f \in [\![\alpha]\!],\, x \in [\![\beta]\!]\}}$$

For $\mathsf{C}\,a ::= (a \to \mathtt{t}) \to \mathtt{t}$, we have an Applicative for scope/continuations:

$$\underbrace{\rho\, x := \lambda k.\, k\, x}_{\text{no correspondent in standard theories}} \qquad \underbrace{m \circledast n := \lambda k.\, m(\lambda f.\, n(\lambda x.\, k\,(f\, x)))}_{\text{no correspondent in standard theories}}$$

Sample derivations: *everyone spoke* (cf. *she₀ spoke*, *a linguist spoke*)

$\mathtt{C\,t}$
$\lambda k.\,\mathrm{eo}\,(\lambda x.\,k\,(\mathrm{spoke}\,x))$

$\mathtt{C\,e}$ — eo

$\mathtt{C\,e} \to \mathtt{C\,t}$ — $\lambda n.\lambda k.\,n\,(\lambda x.\,k\,(\mathrm{spoke}\,x))$

$\circledast$

$\mathtt{C\,(e \to t)}$
$\lambda k.\,k\,\mathrm{spoke}$

$\uparrow$

$\mathtt{e \to t}$
spoke

$\mathtt{G\,t}$
$\lambda g.\,\mathrm{spoke}\,g_0$

$\mathtt{G\,e}$ — $\lambda g.\,g_0$

$\mathtt{G\,e} \to \mathtt{G\,t}$ — $\lambda n.\lambda g.\,\mathrm{spoke}\,(n\,g)$

$\circledast$

$\mathtt{G\,(e \to t)}$
$\lambda g.\,\mathrm{spoke}$

ρ

$\mathtt{e \to t}$
spoke

$\mathtt{S\,t}$
$\{\mathrm{spoke}\,x \mid x \in \mathrm{ling}\}$

$\mathtt{S\,e}$ — $\{x \mid x \in \mathrm{ling}\}$

$\mathtt{S\,e} \to \mathtt{S\,t}$ — $\lambda n.\{\mathrm{spoke}\,x \mid x \in n\}$

$\circledast$

$\mathtt{S\,(e \to t)}$
$\{\mathrm{spoke}\}$

ρ

$\mathtt{e \to t}$
spoke

The $\circledast$ intuition

Across all these cases, we observe a common pattern: ρ and $\circledast$ allow us to "pretend" as if we weren't dealing with fancy things, for the purposes of composing the basic function-argument structure of the sentence.

When I have, e.g., a function $f : a \to b$ and a fancy argument $m : F\,a$, ρ and $\circledast$ allow me to "apply f inside the F" to deliver an $F\,b$.

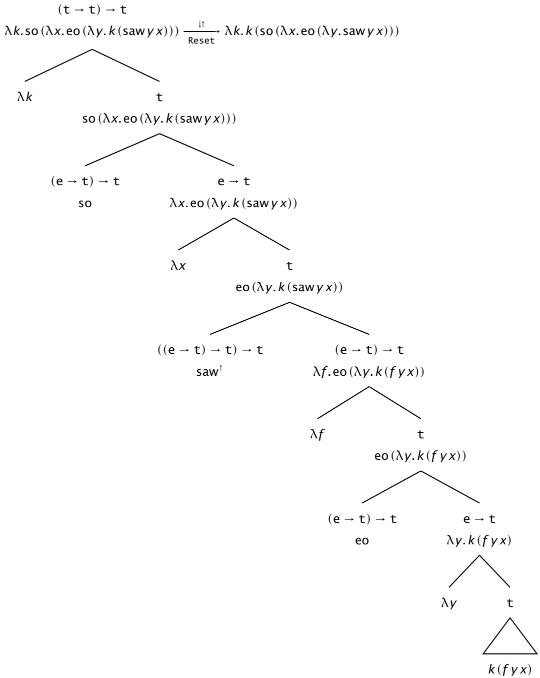
Ending derivations and Reset

This suggests that we need a way to end derivations, and that derivations must be obligatorily concluded at certain points (e.g., at tensed clauses).

$$m^{\downarrow} = m\,(\lambda p.\, p)$$

An example of what this delivers for the present example:

$$(\lambda k.\, \neg\exists x.\, k\,(\text{came}\, x))\,(\lambda p.\, p) = \neg\exists x.\, \text{came}\, x$$



(t → t) → t
λk. so (λx. eo (λy. k (saw y x))) ↓↑ Reset → λk. k (so (λx. eo (λy. saw y x)))
λk
t
so (λx. eo (λy. k (saw y x)))
(e → t) → t
so
e → t
λx. eo (λy. k (saw y x))
λx
t
eo (λy. k (saw y x))
((e → t) → t) → t
saw↑
(e → t) → t
λf. eo (λy. k (f y x))
λf
t
eo (λy. k (f y x))
(e → t) → t
eo
e → t
λy. k (f y x)
λy
t
k (f y x)

Towers

The tower notation

Continuized derivations are easier to appreciate if we help outselves to an ingenious bit of notation known as **towers** (Barker & Shan 2008):

$$\lambda k.\, f\,[k\,x] \longrightarrow \frac{f\,[\;]}{x}$$

A few examples of how this works:

$$\lambda k.\, k\,\mathsf{m} \longrightarrow \frac{[\;]}{\mathsf{m}} \qquad \lambda k.\, k\,\mathsf{saw} \longrightarrow \frac{[\;]}{\mathsf{saw}} \qquad \lambda k.\forall y.\, k\,y \longrightarrow \frac{\forall y.[\;]}{y}$$

Combining towers with $\circledast$

Likewise, $\circledast$ can be naturally re-expressed in the tower notation:

$$\frac{A[\;]}{f} \circledast \frac{B[\;]}{x} \longrightarrow \frac{A[B[\;]]}{f\,x}$$

Notice that we give the function priority over the argument:

$$\left(\frac{[\;]}{\text{saw}}\;\frac{\forall y.[\;]}{y}\right)\frac{\exists x.[\;]}{x} \longrightarrow \frac{\forall y.\exists x.[\;]}{\text{saw}\,y\,x}$$

The result's equivalent to what we'd derive using a linear notation:

$$\lambda k.\forall y.\exists x.\,k\,(\text{saw}\,y\,x)$$

Evaluation order

Shan & Barker (2006) make an interesting, and important suggestion. Instead of always giving *functions* priority. Why not always give *the thing on the left* priority?

$$\underbrace{\frac{A[\ ]}{x}}_{\text{l-exp}}\ \underbrace{\frac{B[\ ]}{y}}_{\text{r-exp}} \longrightarrow \frac{A[B[\ ]]}{x\,y \text{ or } y\,x\text{, whichever's defined}}$$

The "effects" on the top levels are consistently threaded in their *linear* order, regardless of whether we're doing forward or backward application on the bottom.

Shan & Barker refer to this as "left-to-right evaluation". It corresponds to a hypothesis that semantic evaluation has a built-in linear bias.

Somebody saw everybody

Now we can re-present the derivation of *somebody saw everybody*:

$$\frac{\exists x.[\ ]}{x}\left(\frac{[\ ]}{\text{saw}}\ \frac{\forall y.[\ ]}{y}\right) \longrightarrow \frac{\exists x.\forall y.[\ ]}{\text{saw}\,y\,x}$$

This time, we derive *surface scope*. We are no longer giving exclusive priority to the function, but to things on the left.

This seems like a better situation than the one we were in before — we bias surface-scope interpretations, just like speakers/hearers do. And the *explanation* for this bias is intuitive: people evaluate expressions in the order they are heard.

Varieties of $\uparrow$

Because $\uparrow$ can apply to anything, it can also apply to towers:

$$\uparrow \frac{\forall y.[\ ]}{y} \longrightarrow \frac{\color{blue}{[\ ]}}{\dfrac{\forall y.[\ ]}{y}}$$

Less obviously, but no less validly, we are also free to apply $\uparrow$ inside a tower:

$$\frac{[\ ]}{\uparrow}\ \frac{\forall y.[\ ]}{y} \longrightarrow \frac{\forall y.[\ ]}{y^{\uparrow}} \longrightarrow \frac{\forall y.[\ ]}{\dfrac{\color{blue}{[\ ]}}{y}}$$

Big tower combination: composing C with C

$$\frac{A[\;]}{\dfrac{C[\;]}{f}}\ \frac{B[\;]}{\dfrac{D[\;]}{x}} \longrightarrow \frac{A[B[\;]]}{\dfrac{C[D[\;]]}{f\,x}}$$

In fact, this rule follows directly from applying ⊛ *inside* the function tower:

$$\left(\frac{[\;]}{\circledast}\ \frac{A[\;]}{\dfrac{C[\;]}{f}}\right)\frac{B[\;]}{\dfrac{D[\;]}{x}} = \frac{A[B[\;]]}{\dfrac{C[D[\;]]}{f\,x}}$$

In terms of linear notation, three-level tower combination is quite a beast:

$$M \circledast\!\circledast N = \lambda k.\, M(\lambda m.\, N(\lambda n.\, k(\underbrace{m \circledast n}_{\text{instead of combining } m \text{ and } n \text{ by application, combine them by } \circledast})))$$

Inverse scope

And that is all we need to account for inverse scope!

$$\dfrac{[\ ]}{\dfrac{\exists x.[\ ]}{x}} \left(\dfrac{[\ ]}{\dfrac{[\ ]}{\text{saw}}} \; \dfrac{\forall y.[\ ]}{\dfrac{[\ ]}{y}} \right) \longrightarrow \dfrac{\forall y.[\ ]}{\dfrac{\exists x.[\ ]}{\text{saw}\, y\, x}}$$

Lowering

The last piece is re-casting lowering (↓) in terms of towers. Like both ↑ and ⊛, it works automatically for towers of arbitary heights.

$$\frac{f[\;]}{x} \longrightarrow f[x] \qquad\qquad \cfrac{f[\;]}{\cfrac{g[\;]}{x}} \longrightarrow \frac{f[\;]}{g[x]} \longrightarrow f[g[x]]$$

Lowering our inverse-scope derivation:

$$\cfrac{\forall y.[\;]}{\cfrac{\exists x.[\;]}{\mathrm{saw}\,y\,x}} \longrightarrow \forall y.\exists x.\mathrm{saw}\,y\,x$$

Binding

Pronouns

Recall that Jacobson (1999) treats pronouns as *identity functions* on individuals.

Barker & Shan (2014) treat pronouns as identity functions on *continuations*.

$$[\![\text{she}]\!] = \underbrace{\lambda k.\, k}_{(e \to t) \to e \to t}$$

This expression is equivalent in the lambda calculus to the following:

$$\lambda k. \lambda x.\, k\, x$$

Applying the tower notation, we end up with the following:

$$\frac{\lambda x.[\ ]}{x}$$

Basic derivations: *John saw her*, *everybody saw her*

$$\frac{[\;]}{\mathrm{j}}\left(\frac{[\;]}{\mathrm{saw}}\;\frac{\lambda y.[\;]}{y}\right) \longrightarrow \frac{\lambda y.[\;]}{\mathrm{saw}\,y\,\mathrm{j}} \longrightarrow \lambda y.\mathrm{saw}\,y\,\mathrm{j}$$

$$\begin{array}{c}[\;]\\\hline \forall x.[\;]\\\hline x\end{array}\left(\begin{array}{c}[\;]\\\hline [\;]\\\hline \mathrm{saw}\end{array}\;\begin{array}{c}\lambda y.[\;]\\\hline [\;]\\\hline y\end{array}\right) \longrightarrow \begin{array}{c}\lambda y.[\;]\\\hline \forall x.[\;]\\\hline \mathrm{saw}\,y\,x\end{array} \longrightarrow \lambda y.\forall x.\mathrm{saw}\,y\,x$$

Binding and crossover

Anybody needs an operation that allows pronouns to be bound. In standard accounts, this is Predicate Abstraction. In Jacobson's semantics, it's the **Z** rule. Here, the relevant operation *duplicates* an expression's value above the line:

$$\frac{[\ ]}{\mathrm{m}} \longrightarrow \frac{[\ ]\mathrm{m}}{\mathrm{m}} \qquad\qquad \frac{\forall x.[\ ]}{x} \longrightarrow \frac{\forall x.[\ ]x}{x}$$

What this actually amounts to denotationally is the following:

$$m^{\triangleright} = \lambda k.\, m\,(\lambda x.\, k\,x\,x)$$

Notice that without the extra x, this would just be an identity mapping:

$$\lambda k.\, m\,(\lambda x.\, k\,x) = \lambda k.\, m\,k = m$$

Sample derivation: *everybody saw his mom*

$$\frac{\forall x.[\,]\,x}{x}\left(\frac{[\,]}{\text{saw}}\left(\frac{[\,]}{\text{mom}}\;\frac{\lambda y.[\,]}{y}\right)\right) \longrightarrow \frac{\forall x.(\lambda y.[\,])\,x}{\text{saw}\,(\text{mom}\,y)\,x} \longrightarrow \frac{\forall x.[\,]}{\text{saw}\,(\text{mom}\,x)\,x}$$

The L-R bias pipes the quantifier's "extra argument" to the pronoun. Lowering:

$$\forall x.\,\text{saw}\,(\text{mom}\,x)\,x$$

All without any assignment functions!

Types for pronouns

Do pronouns have type $\mathsf{C}\,a ::= (a \to \mathtt{t}) \to \mathtt{t}$ (for some a)?

Types for pronouns

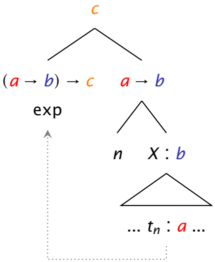
Do pronouns have type $C\,a ::= (a \to \mathtt{t}) \to \mathtt{t}$ (for some a)?

They do not! One concrete type for a pronoun is $(\mathsf{e} \to \mathtt{t}) \to \mathsf{e} \to \mathtt{t}$, which is not of the form $(a \to \mathtt{t}) \to \mathtt{t}$. More generally, we can assign pronouns the following type:

$$(\mathsf{e} \to a) \to \mathsf{e} \to a$$

Generalized types

The most general type for something that takes scope is $(a \to b) \to c$:



Here, exp functions "locally" as something of type a, and takes scope over something of type b, ultimately delivering something of type c:

- Quantifiers: locally e, scope over t, deliver t
- Pronouns: locally e, scope over a, deliver e $\to a$

Towers for types

Just like there's a useful tower notation for continuized (scope-y) values, there's a tower notation for continuized (scope-y) types:

$$\frac{c \,\big|\, b}{a} ::= (a \to b) \to c$$

Here are examples of tower-types for a generalized quantifier, a pronoun, and a $\triangleright$-shifted generalized quantifier:

$$\frac{\mathtt{t} \,\big|\, \mathtt{t}}{\mathtt{e}} \qquad \frac{\mathtt{e} \to a \,\big|\, a}{\mathtt{e}} \qquad \frac{\mathtt{t} \,\big|\, \mathtt{e} \to \mathtt{t}}{\mathtt{e}}$$

Combination with generalized types

Generalizing the types does not change the basic story about how composition works. Composition is still via ρ and $\circledast$, but we are no longer artifically restricting ourselves to scoping over and returning something of type t, as we have done.

For example, the action of ρ can be seen in terms of tower types as:

$$a \longrightarrow \begin{array}{c|c} b & b \\ \hline \multicolumn{2}{c}{a} \end{array}$$

And the action of $\circledast$ can likewise be seen in terms of tower types as:

$$\begin{array}{c|c} \color{blue}{c} & \color{orange}{d} \\ \hline \multicolumn{2}{c}{a \to b} \end{array} \;\; \begin{array}{c|c} \color{orange}{d} & \color{red}{e} \\ \hline \multicolumn{2}{c}{a} \end{array} \longrightarrow \begin{array}{c|c} \color{blue}{c} & \color{red}{e} \\ \hline \multicolumn{2}{c}{b} \end{array}$$

Left expects to scope over a d, and so right must return a d.

Basic derivation: *he saw her*

Here is a derivation of *he saw her* in terms of value-towers:

$$\frac{\lambda x.[\;]}{x}\left(\frac{[\;]}{\mathsf{saw}}\;\frac{\lambda y.[\;]}{y}\right) \longrightarrow \frac{\lambda x.\lambda y.[\;]}{\mathsf{saw}\,y\,x} \longrightarrow \lambda x.\lambda y.\mathsf{saw}\,y\,x$$

Sketching the type-towers in the derivation is revealing, as well:

$$\frac{\mathtt{e}\to\mathtt{e}\to\mathtt{t}\,\big|\,\mathtt{e}\to\mathtt{t}}{\mathtt{e}}\left(\frac{\mathtt{e}\to\mathtt{t}\,\big|\,\mathtt{e}\to\mathtt{t}}{\mathtt{e}\to\mathtt{e}\to\mathtt{t}}\;\frac{\mathtt{e}\to\mathtt{t}\,\big|\,\mathtt{t}}{\mathtt{e}}\right) \longrightarrow \frac{\mathtt{e}\to\mathtt{e}\to\mathtt{t}\,\big|\,\mathtt{t}}{\mathtt{t}} \longrightarrow \mathtt{e}\to\mathtt{e}\to\mathtt{t}$$

This reveals something important about $\downarrow$...

Basic derivation: *he saw her*

Here is a derivation of *he saw her* in terms of value-towers:

$$\frac{\lambda x.[\ ]}{x}\left(\frac{[\ ]}{\mathtt{saw}}\;\frac{\lambda y.[\ ]}{y}\right) \longrightarrow \frac{\lambda x.\lambda y.[\ ]}{\mathtt{saw}\,y\,x} \longrightarrow \lambda x.\lambda y.\mathtt{saw}\,y\,x$$

Sketching the type-towers in the derivation is revealing, as well:

$$\frac{\mathtt{e}\to\mathtt{e}\to\mathtt{t}\,\big|\,\mathtt{e}\to\mathtt{t}}{\mathtt{e}}\left(\frac{\mathtt{e}\to\mathtt{t}\,\big|\,\mathtt{e}\to\mathtt{t}}{\mathtt{e}\to\mathtt{e}\to\mathtt{t}}\;\frac{\mathtt{e}\to\mathtt{t}\,\big|\,\mathtt{t}}{\mathtt{e}}\right) \longrightarrow \frac{\mathtt{e}\to\mathtt{e}\to\mathtt{t}\,\big|\,\mathtt{t}}{\mathtt{t}} \longrightarrow \mathtt{e}\to\mathtt{e}\to\mathtt{t}$$

This reveals something important about $\downarrow$... it requires matching types!

$$\frac{b\,\big|\,a}{a} \longrightarrow b$$

Binding with types: *every boy$_i$ likes his$_i$ mom*

$$\frac{\mathtt{t} \mid \mathtt{e} \to \mathtt{t}}{\mathtt{e}} \left(\frac{\mathtt{e} \to \mathtt{t} \mid \mathtt{e} \to \mathtt{t}}{\mathtt{e} \to \mathtt{e} \to \mathtt{t}} \left(\frac{\mathtt{e} \to \mathtt{t} \mid \mathtt{e} \to \mathtt{t}}{\mathtt{e} \to \mathtt{e}} \; \frac{\mathtt{e} \to \mathtt{t} \mid \mathtt{t}}{\mathtt{e}} \right) \right) \longrightarrow \frac{\mathtt{t} \mid \mathtt{t}}{\mathtt{t}}$$

The out-to-bind quantifier and the out-to-be-bound pronoun exercise a magnetic pull on each other, one which linear-biased $\circledast$ brings to fruition.

Crossover

Quantifiers cannot bind pronouns that precede them, even though the quantifier can scope over another quantifier in that position:

1. A doctor examined every patient.
2. *His$_i$ mother examined every patient$_i$.
3. *He$_i$ examined every patient$_i$.
4. *A friend of his$_i$ examined every patient$_i$.

How is crossover normally enforced?

Crossover

Quantifiers cannot bind pronouns that precede them, even though the quantifier can scope over another quantifier in that position:

1. A doctor examined every patient.
2. *His$_i$ mother examined every patient$_i$.
3. *He$_i$ examined every patient$_i$.
4. *A friend of his$_i$ examined every patient$_i$.

How is crossover normally enforced? There are a number of approaches, but one way or another, they prevent a quantifier from binding a pronoun by moving over it.

However, this is problematic. Other examples suggest that the right account of crossover is not in terms of *hierarchy* but rather in terms of *linear order*.

Crossover

Quantifiers cannot bind pronouns that precede them, even though the quantifier can scope over another quantifier in that position:

1. A doctor examined every patient.
2. *His$_i$ mother examined every patient$_i$.
3. *He$_i$ examined every patient$_i$.
4. *A friend of his$_i$ examined every patient$_i$.

How is crossover normally enforced? There are a number of approaches, but one way or another, they prevent a quantifier from binding a pronoun by moving over it.

However, this is problematic. Other examples suggest that the right account of crossover is not in terms of *hierarchy* but rather in terms of *linear order*.

5. [Someone from every city$_i$] likes it$_i$.

Explaining crossover: trying surface scope

Let's try and construct a surface-scope derivation for *he$_i$ examined every patient$_i$*.

$$\frac{\mathtt{e \to t} \,\big|\, \mathtt{t}}{\mathtt{e}} \left(\frac{\mathtt{t} \,\big|\, \mathtt{t}}{\mathtt{e \to e \to t}} \; \frac{\mathtt{t} \,\big|\, \mathtt{e \to t}}{\mathtt{e}} \right) \longrightarrow \frac{\mathtt{e \to t} \,\big|\, \mathtt{e \to t}}{\mathtt{t}}$$

There is no way to conclude the derivation. Barker & Shan (2008) put it well:

> *At best, the pronoun mournfully looks left for an antecedent, while the binder looks right for a pronoun to bind.*

Explaining crossover: trying inverse scope

$$\dfrac{\texttt{t} \mid \texttt{t}}{\dfrac{\texttt{e} \to \texttt{t} \mid \texttt{t}}{\texttt{e}}} \left(\dfrac{\texttt{t} \mid \texttt{t}}{\dfrac{\texttt{t} \mid \texttt{t}}{\texttt{e} \to \texttt{e} \to \texttt{t}}} \; \dfrac{\texttt{t} \mid \texttt{e} \to \texttt{t}}{\dfrac{\texttt{t} \mid \texttt{t}}{\texttt{e}}} \right) \longrightarrow \dfrac{\texttt{t} \mid \texttt{e} \to \texttt{t}}{\dfrac{\texttt{e} \to \texttt{t} \mid \texttt{t}}{\texttt{t}}} \longrightarrow \dfrac{\texttt{t} \mid \texttt{e} \to \texttt{t}}{\texttt{e} \to \texttt{t}}$$

Can we conclude this derivation? If we stick to our old version of ↓, we could since the bottom and top-right types match. But we can *place a restriction on* ↓:

Explaining crossover: trying inverse scope

$$\begin{array}{c} \mathtt{t} \mid \mathtt{t} \\ \hline \mathtt{e} \to \mathtt{t} \mid \mathtt{t} \\ \hline \mathtt{e} \end{array} \left(\begin{array}{c} \mathtt{t} \mid \mathtt{t} \\ \hline \mathtt{t} \mid \mathtt{t} \\ \hline \mathtt{e} \to \mathtt{e} \to \mathtt{t} \end{array} \; \begin{array}{c} \mathtt{t} \mid \mathtt{e} \to \mathtt{t} \\ \hline \mathtt{t} \mid \mathtt{t} \\ \hline \mathtt{e} \end{array} \right) \longrightarrow \begin{array}{c} \mathtt{t} \mid \mathtt{e} \to \mathtt{t} \\ \hline \mathtt{e} \to \mathtt{t} \mid \mathtt{t} \\ \hline \mathtt{t} \end{array} \longrightarrow \begin{array}{c} \mathtt{t} \mid \mathtt{e} \to \mathtt{t} \\ \hline \mathtt{e} \to \mathtt{t} \end{array}$$

Can we conclude this derivation? If we stick to our old version of ↓, we could since the bottom and top-right types match. But we can *place a restriction on* ↓:

$$\begin{array}{c} a \mid \mathtt{t} \\ \hline \mathtt{t} \end{array} \longrightarrow a$$

In other words, you're only allowed to "end" derivations of sentences.

Stepping back

The prohibition on crossover comes from two places: the inherent L-R bias in composition, and a stipulation to the effect that only complete sentences can be $\downarrow$'d.

Taken together, these require a bound pronoun to follow its binder. There seem to be some counterexamples:

6. Himself$_i$, every man$_i$ loves _.
7. Which of his$_i$ relatives does every man$_i$ love _ the most?
8. Unless he$_i$'s been an officer, no man$_i$ can be a gentleman _.

Moral: the right notion is *reconstructed* linear order. Shan & Barker (2006), Barker & Shan (2014) give a full fragment treating reconstruction.

Islands

There is a slight hiccup. If pronouns take scope, and things need to be lowered at islands, how can pronouns scope out of islands?

9. Every philosopher$_i$ thinks [they$_i$'re a genius].

For the [tensed clause] (a scope island), we derive $\lambda x.\,\text{genius}\,x$. We need to convert this back to a tower so it can interact with the $\triangleright$-shifted subject in the desired way.

Moreover, whatever solution we give should be general enough to handle cases like:

10. Every boy$_i$ told every girl$_j$ that [he$_i$ was paired with her$_j$].

By and large, these issues have not been dealt with by continuations-istas. There is one exception (Charlow 2014).

Barker, Chris & Chung-chieh Shan. 2008. Donkey anaphora is in-scope binding. *Semantics and Pragmatics* 1(1). 1–46. https://doi.org/10.3765/sp.1.1.

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

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

Charlow, Simon. 2017. A modular theory of pronouns and binding. In *Proceedings of Logic and Engineering of Natural Language Semantics 14*.

Jacobson, Pauline. 1999. Towards a variable-free semantics. *Linguistics and Philosophy* 22(2). 117–184. https://doi.org/10.1023/A:1005464228727.

Kiselyov, Oleg. 2015. Applicative abstract categorial grammars. In Makoto Kanazawa, Lawrence S. Moss & Valeria de Paiva (eds.), *NLCS'15. Third workshop on natural language and computer science*, vol. 32 (EPiC Series), 29–38.

McBride, Conor & Ross Paterson. 2008. Applicative programming with effects. *Journal of Functional Programming* 18(1). 1–13. https://doi.org/10.1017/S0956796807006326.

Shan, Chung-chieh & Chris Barker. 2006. Explaining crossover and superiority as left-to-right evaluation. *Linguistics and Philosophy* 29(1). 91–134. https://doi.org/10.1007/s10988-005-6580-7.