Focus (and more monadic interaction)

Computational semantics seminar

March 29, 2017

Some basic data

Question-answer congruence

1. **Who came to the party?**
   - 1.1 BILL$_F$ came to the party.
   - 1.2 #Bill came to [the PARTY]$_F$.
   - 1.3 #BILL$_F$ came to [the PARTY]$_F$.

No obvious truth-conditional difference between the three responses. However, only some are felicitous in context.

- The parts of the answer corresponding to the *wh* must be focused.
- Nothing else in the answer can be focused.

Discourse congruence

1. **John came to the party.**
   - 1.1 And BILL$_F$ came to the party.
   - 1.2 #And Bill came to [the PARTY]$_F$.
   - 1.3 #And BILL$_F$ came to [the PARTY]$_F$.

No obvious truth-conditional difference between the three continuations. However, only some are felicitous in context.

- Parts of the continuation contrasting w/the setup must be focused.
- Nothing else in the continuation can be focused.

Association with focus

1. **No focus-sensitive expression:**
   - 1.1 John introduced BILL_F to Sue.
   - 1.2 John introduced Bill to SUE_F.
   - 1.3 John introduced BILL_F to SUE_F.

$\rightsquigarrow$ No clear truth-conditional difference.

2. **A focus-sensitive adverb:**
   - 2.1 John only introduced BILL_F to Sue.
   - 2.2 John only introduced Bill to SUE_F.
   - 2.3 John only introduced BILL_F to SUE_F.

$\rightsquigarrow$ Truth-conditional differences!

Islands

All these phenomena display a characteristic insensitivity to islands:

1. **Which linguist will be mad if [we invite which philosopher]?**

   1.1 CHOMSKY$_F$ will be mad if [we invite QUINE$_F$].
   1.2 #CHOMSKY$_F$ will be mad if [we invite Quine].
   1.3 #CHOMSKY$_F$ will be mad if [WE$_F$ invite QUINE$_F$].

2. **John gets mad when [Sue leaves the lights on].**

   2.1 And SUE$_F$ gets mad when [JOHN$_F$ leaves the lights on]!
   2.2 #And SUE$_F$ gets mad when [John leaves the lights on]!
   2.3 #And SUE$_F$ gets mad when [JOHN$_F$ leaves the lights ON$_F$]!

3. **John only gets mad when [SUE$_F$ leaves the lights on].**

Focus semantics

Two possibilities for focus projection

We'll review two possibilities for treating focus:

- Alternative semantics (Rooth 1985)
- Structured meanings (e.g., Krifka 1991, 2006)

We'll see that they turn on different assumptions about the nature of focus projection, and the syn-sem interface.

- Alternative semantics is pseudo-scopal (or in situ)
- Structured meanings are scope-based

Alternative semantics à la Rooth (1985)

Two interpretation functions: $[\![\cdot]\!]$ and $\{\!\{\cdot\}\!\}$.

- $[\![\cdot]\!]$ is the familiar interpretation function you know and (sometimes) love: functional application, predicate modification, ...
- $\{\!\{\cdot\}\!\}$ is an alternative-semantic interpretation function. It works by point-wise composing *sets* of meanings.

Basic meanings

Meanings for unfocused and focused expressions:

$$[\![\text{Bill}]\!] := \mathbf{b} \qquad \{\!\{\text{Bill}\}\!\} := \{\mathbf{b}\}$$

$$[\![\text{SUE}_F]\!] := \mathbf{s} \qquad \{\!\{\text{SUE}_F\}\!\} := \underbrace{\{x \mid x \in \mathbf{alt_s}\}}_{\text{i.e., } \mathbf{alt_s}}$$

Meanings for binary-branching nodes:

$$[\![X\ Y]\!] := [\![X]\!][\![Y]\!] \qquad \{\!\{X\ Y\}\!\} := \{x\,y \mid x \in \{\!\{X\}\!\}, y \in \{\!\{Y\}\!\}\}$$

An example "calculation":

$$[\![\text{Bill saw SUE}_F]\!] = \mathbf{saw\,s\,b} \qquad \{\!\{\text{Bill saw SUE}_F\}\!\} = \{\mathbf{saw}\,x\,\mathbf{b} \mid x \in \mathbf{alt_s}\}$$

Only

Normal and focus meanings for (propositional) *only*:

$$[\![\text{only } S]\!] := \lambda w.[\![S]\!]w \wedge \forall p \in \{\!\{S\}\!\} : pw \Rightarrow p = [\![S]\!]$$
$$\{\!\{\text{only } S\}\!\} := \{[\![\text{only } S]\!]\}$$

Two important properties:

- Meaning is specified syncategorematically (why?)
- Meaning resets the focus value (why?)

Congruence

Normal and focused meanings for a *focus interpretation* operator:

$$[\![X \sim C]\!] := \begin{cases} [\![X]\!] \text{ if } \{\!\{X\}\!\} \sim C \\ \# \quad \text{otherwise} \end{cases}$$

$$\{\!\{X \sim C\}\!\} := \{[\![X \sim C]\!]\}$$

For question-answer congruence, $\sim$ might resolve to $\supseteq$. For discourse congruence, $\sim$ might resolve to $\ni$.

Like *only*, has two important properties:

- Meaning is specified syncategorematically (why?)
- Meaning resets the focus value (why?)

More direct two-dimensionality

Meanings for unfocused and focused expressions:

$$\begin{aligned} [\![\text{Bill}]\!] &:= (\mathbf{b}, \{\mathbf{b}\}) \\ [\![\text{SUE}_F]\!] &:= (\mathbf{s}, \mathbf{alt_s}) \end{aligned}$$

Meaning for binary-branching nodes:

$$[\![X\ Y]\!] := (\underbrace{\mathbf{fst}[\![X]\!](\mathbf{fst}[\![Y]\!])}_{\text{normal meaning}}, \overbrace{\{x\,y \mid x \in \mathbf{snd}[\![X]\!], y \in \mathbf{snd}[\![Y]\!]\}}^{\text{focus meaning}})$$

This is notationally heavier, but it's a bit clearer about how the underlying compositional machinery is working. Hold it in your mind's eye.

Structured meanings

The Roothian picture of focus projection leaves focused things in situ, using a compositional mechanism to project their alternatives upward.

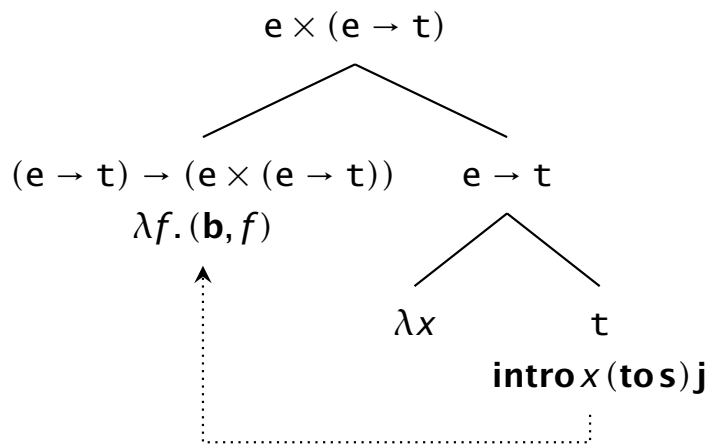
An alternative is to suppose that focused things take scope, and use this mechanism to build different kinds of focus meanings:

$$\textit{John met MARY}_F \rightsquigarrow (\mathbf{m}, \lambda x.\, \mathbf{met}\, x\, \mathbf{j})$$

Basics

Here is one simple way to implement structured meanings:

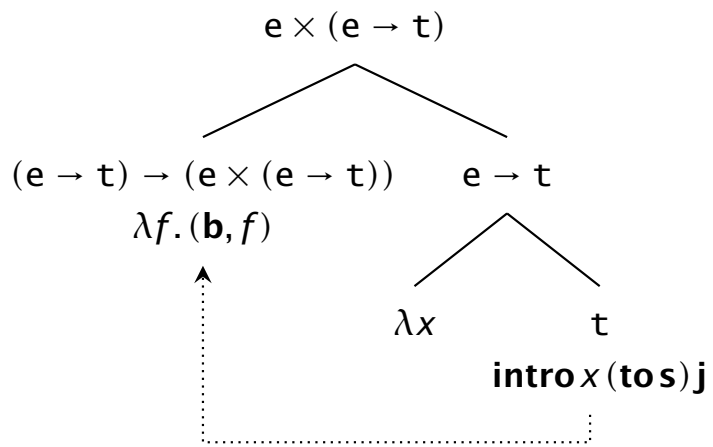
$$\cdot_{\mathbf{F}} :: a \to (a \to b) \to (a \times (a \to b))$$
$$\cdot_{\mathbf{F}} := \lambda x.\lambda f.(x, f)$$



Basics

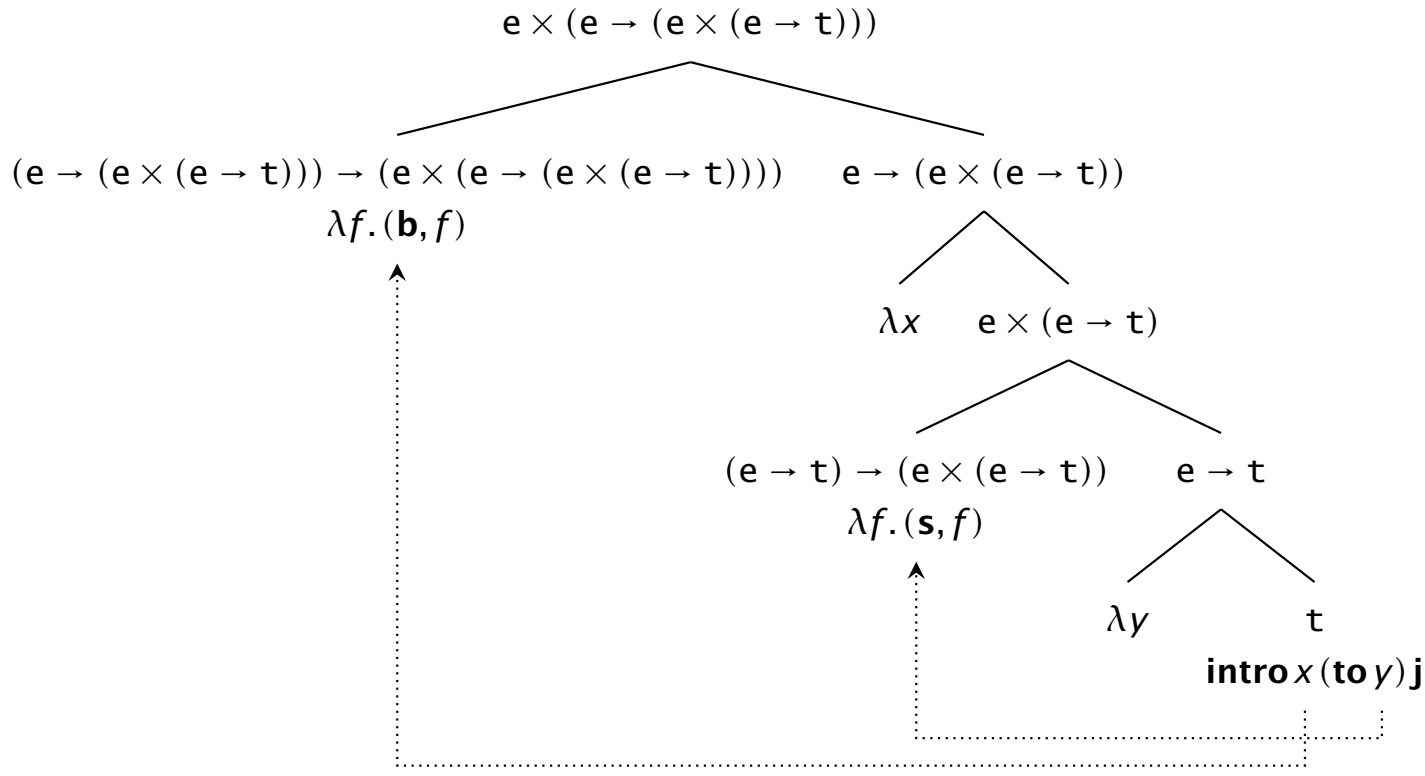
Here is one simple way to implement structured meanings:

$$\cdot_{\mathbf{F}} :: a \to (a \to b) \to (a \times (a \to b))$$
$$\cdot_{\mathbf{F}} := \lambda x.\lambda f.(x, f)$$

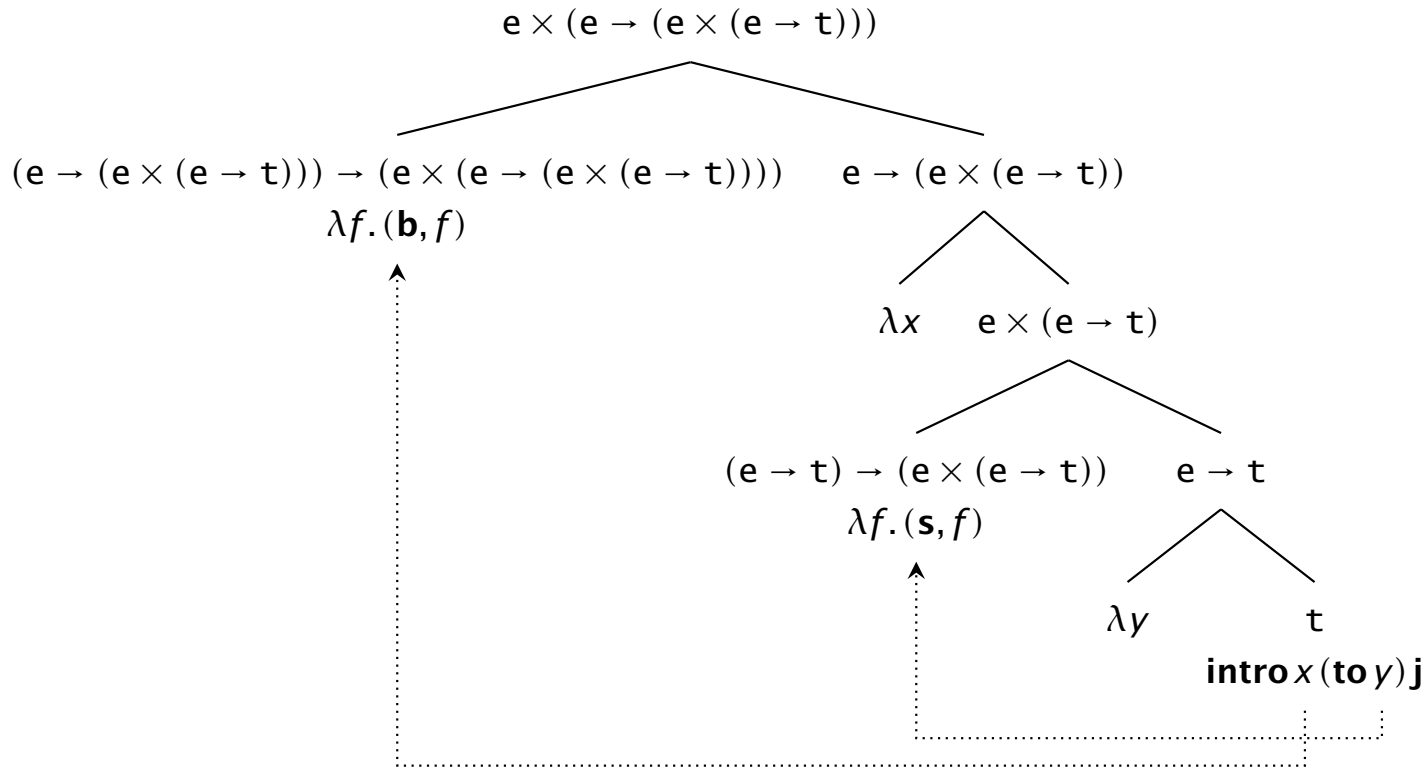


$$(\mathbf{b}, \lambda x.\mathbf{intro}\, x\, (\mathbf{to}\, \mathbf{s})\, \mathbf{j})$$

Multiple focus



Multiple focus



$$(\mathbf{b},\, \lambda x.(\mathbf{s},\, \lambda y.\mathbf{intro}\, x\, (\mathbf{to}\, y)\, \mathbf{j}))$$

[Can be flattened into $((\mathbf{b}, \mathbf{s}), \lambda(x, y).\mathbf{intro}\, x\, (\mathbf{to}\, y)\, \mathbf{j})$]

A familiar dialectic?

The structured meaning approach *looks* inherently scopal, while alternative semantics uses pseudo-scope to project alternatives.

Does this remind you of anything?

A familiar dialectic?

The structured meaning approach *looks* inherently scopal, while alternative semantics uses pseudo-scope to project alternatives.

Does this remind you of anything? Sure, it's precisely the dialectic that characterized scopal approaches to questions (à la Karttunen) from in situ approaches (à la Hamblin).

For alternative semantics: expressive power

Clearly, alternative semantics is less expressive than structured meanings. We can define a mapping from a structured-meaning value into an alternative semantic value, but not vice versa:

$$\textbf{to.alt}\,(x, f) := \{f\,y \mid y \in \textbf{alt}_x\}$$

So there's a real sense in which alternative semantics is the more parsimonious theory. If we can get away with using it, we should.

For alternative semantics: islands

The major success of alternative semantics is that it projects focus alternatives out of islands:

$$\{\!\{\text{John gets mad when [SUE}_F \text{ leaves the lights on]}\}\!\}$$
$$= \{\mathbf{mad.when}\,(\mathbf{leaves.lights.on}\,x)\,\mathbf{j} \mid x \in \mathbf{alt_s}\}$$

Lets adverbial association and congruence checks happen at a distance.

Against alternative semantics: binding

How should we characterize abstraction in the second dimension?

$$\{\!\{\lambda_i\ X\}\!\}^g :=$$

Against alternative semantics: binding

How should we characterize abstraction in the second dimension?

$$\{\!\{\lambda_i\ X\}\!\}^g := \{\underbrace{\lambda x.\ \{\!\{X\}\!\}^{g[i \to x]}}_{\text{already a function into sets!}}\}$$

Precisely the same issue as for alternative semantics. There is no way to fix with this layering of sets and assignments. But this would help:

Against alternative semantics: binding

How should we characterize abstraction in the second dimension?

$$\{\!\{\lambda_i\ X\}\!\}^g := \{\underbrace{\lambda x.\, \{\!\{X\}\!\}^{g[i \to x]}}_{\text{already a function into sets!}}\}$$

Precisely the same issue as for alternative semantics. There is no way to fix with this layering of sets and assignments. But this would help:

$$\{\!\{\lambda_i\ X\}\!\} := \{\lambda g.\lambda x.\, f\, g^{[i \to x]} \mid f \in \{\!\{X\}\!\}\}$$

Something to think about: are there any arguments against this layering? Cases where the *form* of alternatives depends on how variables are set?

For assignment-dependent alternatives

Evidence that alternatives may depend on how pronouns are resolved:

1. Every boy here$_i$ only respects [his$_i$ MOTHER]$_F$.
2. Every boy$_i$ thinks his$_i$ mom is smart.
   Every boy$_i$ also thinks [HE$_i$]$_F$ is smart.

There is some wiggle room for the first example. This intonation pattern is consistent with narrow focus on *MOTHER*. The second example is more telling. Seems to require something like the following:

$$\{\!\{[\text{HE}_i]_F\}\!\}^g = \{x \mid x \in \mathbf{alt}_{g_i}\}$$

But this is precisely the layering alternative semantics can't abide!

Selectivity

Krifka (1991) argues that structured meanings are a natural fit for multiple-association examples like the following:

1. John only introduced BILL_F to Sue.
   He even only introduced BILL_F to MARY_F.

2. John only introduced BILL_F to Sue.
   He also only introduced BILL_F to MARY_F.

How come? Krifka argues that the requirement for focus-sensitive operators to reset alternatives makes multiple association impossible.

- Doesn't go through. Alternative semanticists can always *scope* one of the foci out of the scope of one of the focus-sensitive operators.

Selectivity, redux

But other cases do prove problematic for alternative semantics:

1. [John only gripes when [MARY$_F$ leaves the lights on]]$_C$
   And [MARY$_F$ only gripes when [[JOHN$_F$]$_F$ leaves the lights on]]$_{\sim C}$

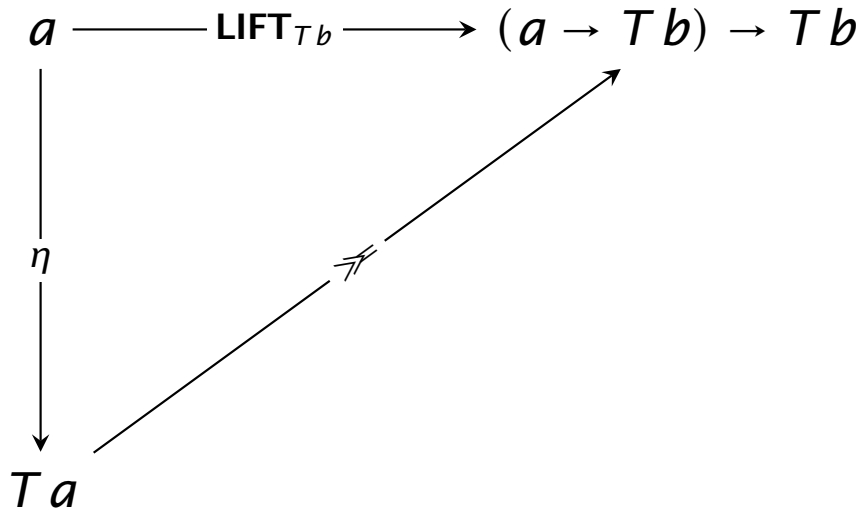
What's wrong here? *JOHN* needs two F-marks, since it associates both with *only* and with $\sim$. But because *JOHN* is embedded in an island, all of its F-marks are going to get swallowed up by *only*!

But notice that this is *also* a problem for structured meanings.

Monadic focus

LOL

A **monad** is a type constructor T associated with η and $\gg\!=$ operations, that makes the following diagram *commute* and satisfies two further laws.



$$m \gg\!= \eta = m \qquad (\textit{Right Id})$$

$$(m \gg\!= f) \gg\!= g = m \gg\!= \lambda x.\, f\, x \gg\!= g \qquad (\textit{Assoc})$$

Type constructor

Given some type *a*, how should we characterize the type of a **focused** *a*?

F *a* ::=

Type constructor

Given some type a, how should we characterize the type of a **focused** a?

$$\mathsf{F}\,a ::= (a, \mathsf{S}\,a)$$

That is: a focused a is a pair of an a and a set of a's. Makes sense!

η

What is the most trivial possible way to turn anything of type a into something of type $\mathsf{F}\,a ::= (a, \mathsf{S}\,a)$?

$$\eta\, x :=$$

η

What is the most trivial possible way to turn anything of type a into something of type $\mathsf{F}\,a ::= (a, \mathsf{S}\,a)$?

$$\eta\, x := (x, \{x\})$$

How can we combine $(x, X) :: \mathrm{F}\, a$ with $f :: a \to \mathrm{F}\, b$ to make an $\mathrm{F}\, b$?

$$(x, Y) \gg\!= f :=$$

$\gg=$

How can we combine $(x, X) :: \mathrm{F}\, a$ with $f :: a \to \mathrm{F}\, b$ to make an $\mathrm{F}\, b$?

$$(x, Y) \gg= f := (\mathbf{fst}\,(f\, x), \bigcup_{y \in Y} \mathbf{snd}\,(f\, y))$$

In other words, do functional application in the first dimension, and do alternative-friendly combination in the second.

Shan (2002) calls this the “Pointed Powerset” monad. Outside of his work and Charlow (2014), it doesn’t seem to have been used anywhere.

Formal note: Focus as a product monad

For any monadic type constructors T and T, $T \times T$ is a monad!

$$\eta\, x := (\eta\, x, \eta\, x) \qquad (m, n) \gg\!= f := (m \gg\!= \mathbf{fst} \circ f, n \gg\!= \mathbf{snd} \circ f)$$

The **Focus** monad is the product of

Formal note: Focus as a product monad

For any monadic type constructors T and T, $T \times T$ is a monad!

$$\eta\, x := (\eta\, x, \eta\, x) \qquad (m, n) \gg\!= f := (m \gg\!= \mathbf{fst} \circ f, n \gg\!= \mathbf{snd} \circ f)$$

The **Focus** monad is the product of the **Set** monad (the monad for alternatives) and

Formal note: Focus as a product monad

For any monadic type constructors T and T, $T \times T$ is a monad!

$$\eta\, x := (\eta\, x, \eta\, x) \qquad (m, n) \gg\!= f := (m \gg\!= \mathbf{fst} \circ f, n \gg\!= \mathbf{snd} \circ f)$$

The **Focus** monad is the product of the **Set** monad (the monad for alternatives) and the **Identity** monad (the monad that does nothing)!

$$\begin{aligned} \eta\, x &:= x & \eta\, x &:= \{x\} \\ m \gg\!= f &:= f\, m & m \gg\!= f &:= \bigcup_{x \in m} f\, x \end{aligned}$$

Some lexical entries

Our grammar stays largely the same, but we provide in addition a meaning for F-marking:

Some lexical entries

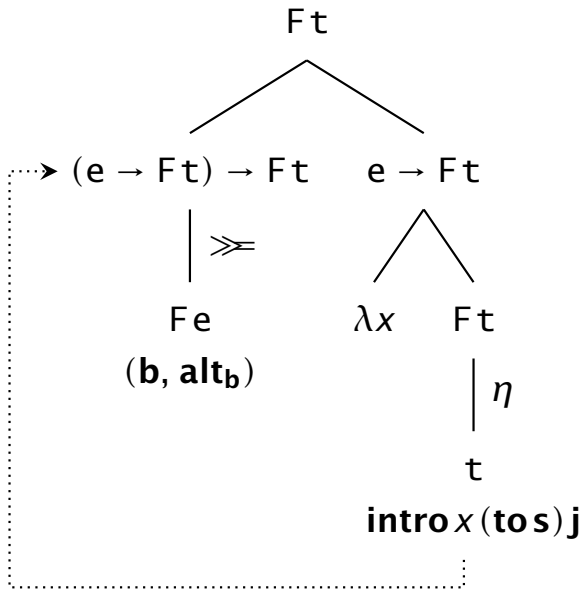
Our grammar stays largely the same, but we provide in addition a meaning for F-marking:

$$\cdot_{\mathsf{F}} :: a \to \mathsf{F}\, a$$
$$\cdot_{\mathsf{F}} := \lambda x.\, (x, \mathbf{alt}_x)$$

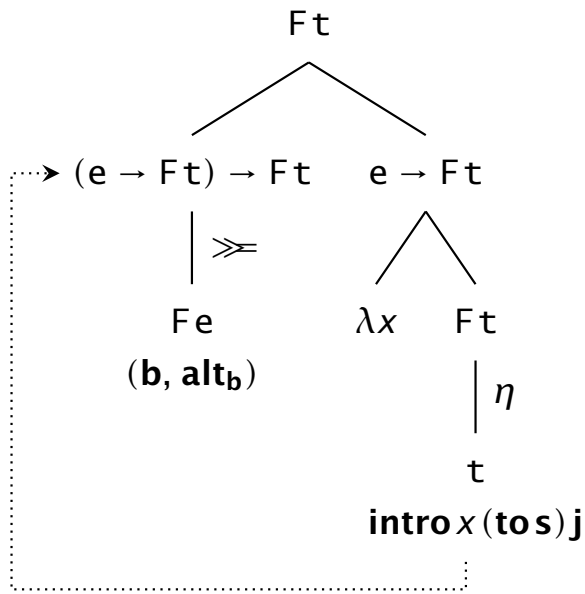
[Same type as η, but a different function.]

And *that's it*. No need to specify $\{\!\{\cdot\}\!\}$ for each terminal node, or say how $\{\!\{\cdot\}\!\}$ composes two meanings. As always, our monadic grammar will be built on simple functional application, and **scope**.

Basic derivation

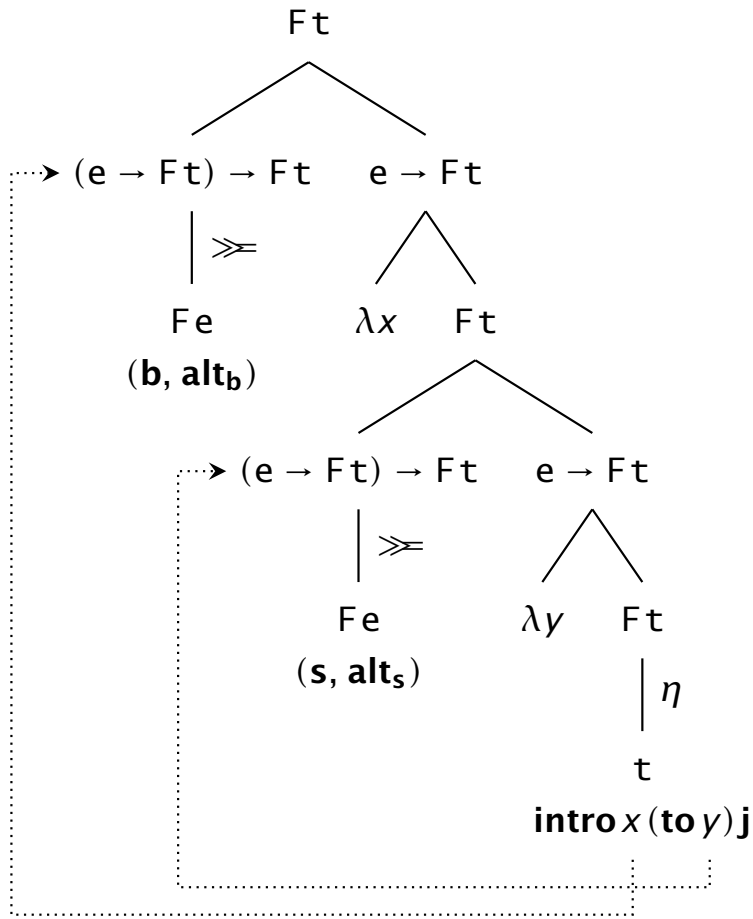


Basic derivation

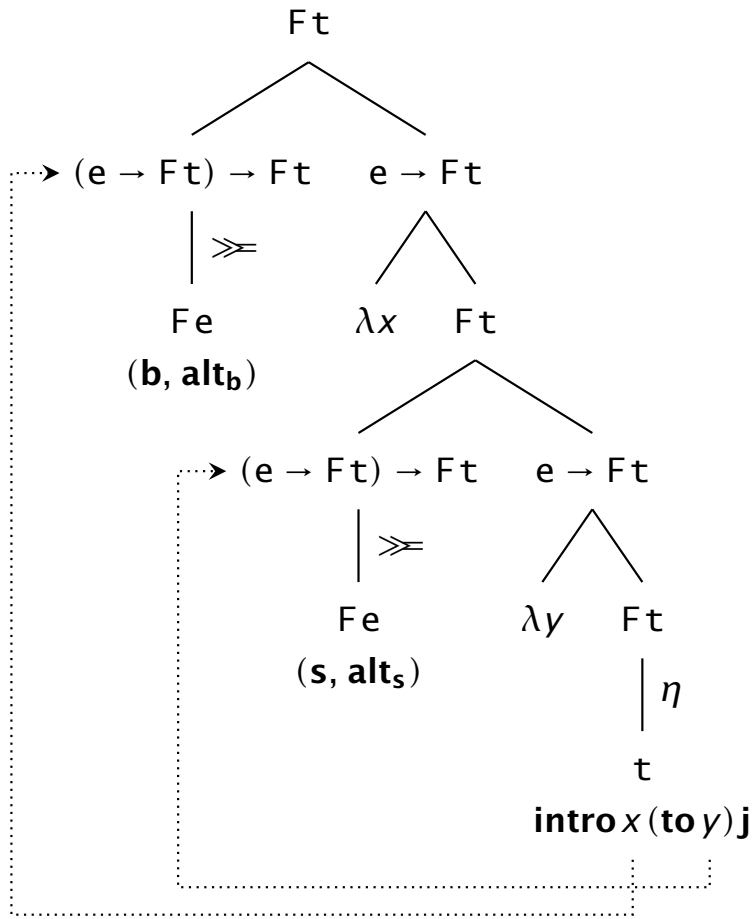


$$(\mathbf{intro}\,\mathbf{b}\,(\mathbf{to}\,\mathbf{s})\,\mathbf{j},\ \{\mathbf{intro}\,x\,(\mathbf{to}\,\mathbf{s})\,\mathbf{j} \mid x \in \mathbf{alt_b}\})$$

Multiple foci

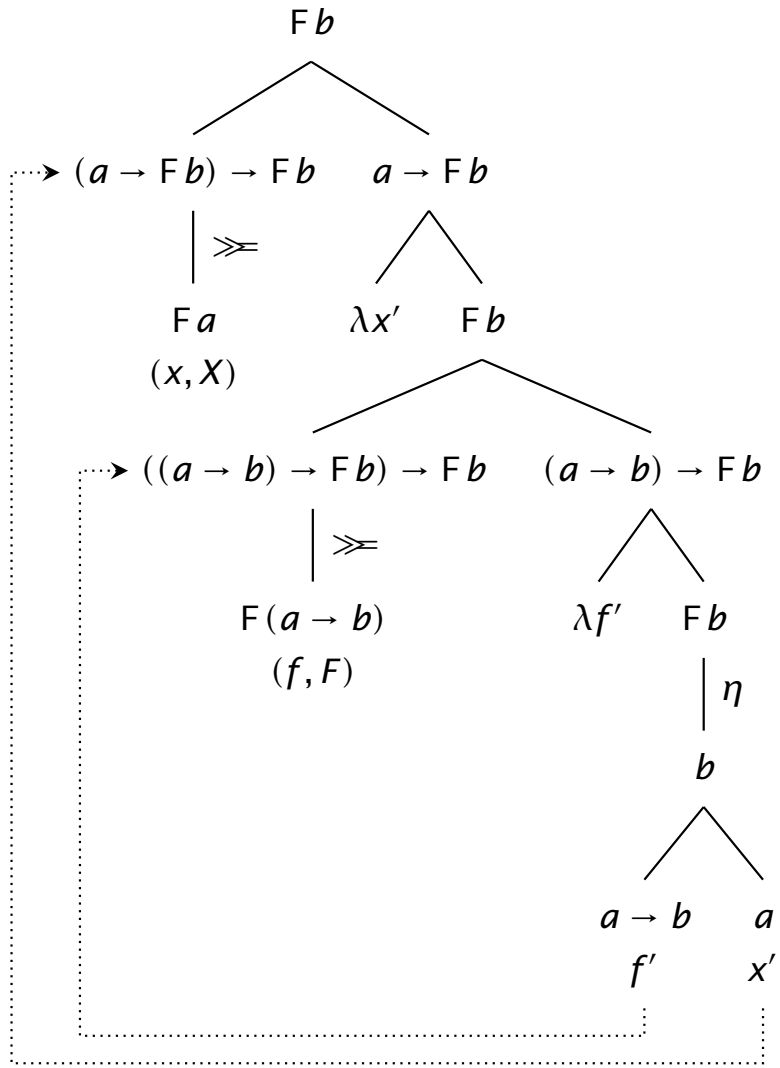


Multiple foci

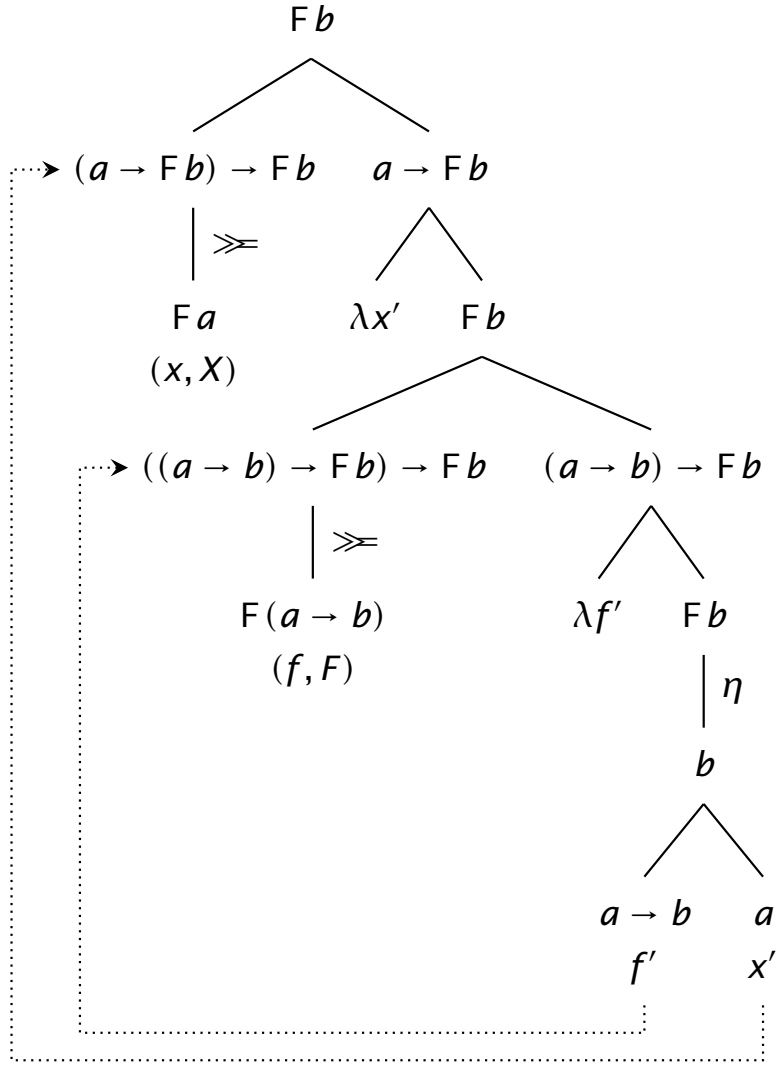


$$(\mathbf{intro}\,\mathbf{b}\,(\mathbf{to}\,\mathbf{s})\,\mathbf{j},\ \{\mathbf{intro}\,x\,(\mathbf{to}\,y)\,\mathbf{j} \mid x \in \mathbf{alt_b}, y \in \mathbf{alt_s}\})$$

Monadic application — look familiar?



Monadic application — look familiar?



$$(f\,x, \{f'\,x' \mid x' \in X, f' \in F\})$$

Meaning for *only*

Takes an $\mathtt{F}\,\mathtt{t}$, returns the proposition that the "normal" meaning is the only true thing in the set of alternatives:

$$\begin{aligned} &\textbf{only} && :: \mathtt{F}\,\mathtt{t} \to \mathtt{t} \\ &\textbf{only}\,(p, A) && := \end{aligned}$$

Meaning for *only*

Takes an $\mathtt{F}\,\mathtt{t}$, returns the proposition that the "normal" meaning is the only true thing in the set of alternatives:

$$\begin{aligned}&\textbf{only} && :: \mathtt{F}\,\mathtt{t} \to \mathtt{t}\\ &\textbf{only}\,(p, A) && := \lambda w.\, p\,w \wedge \forall\, q \in A : q\,w \Rightarrow q = p\end{aligned}$$

Unlike the corresponding Roothian entry, is categorematicizable. Why?

Focus interpretation

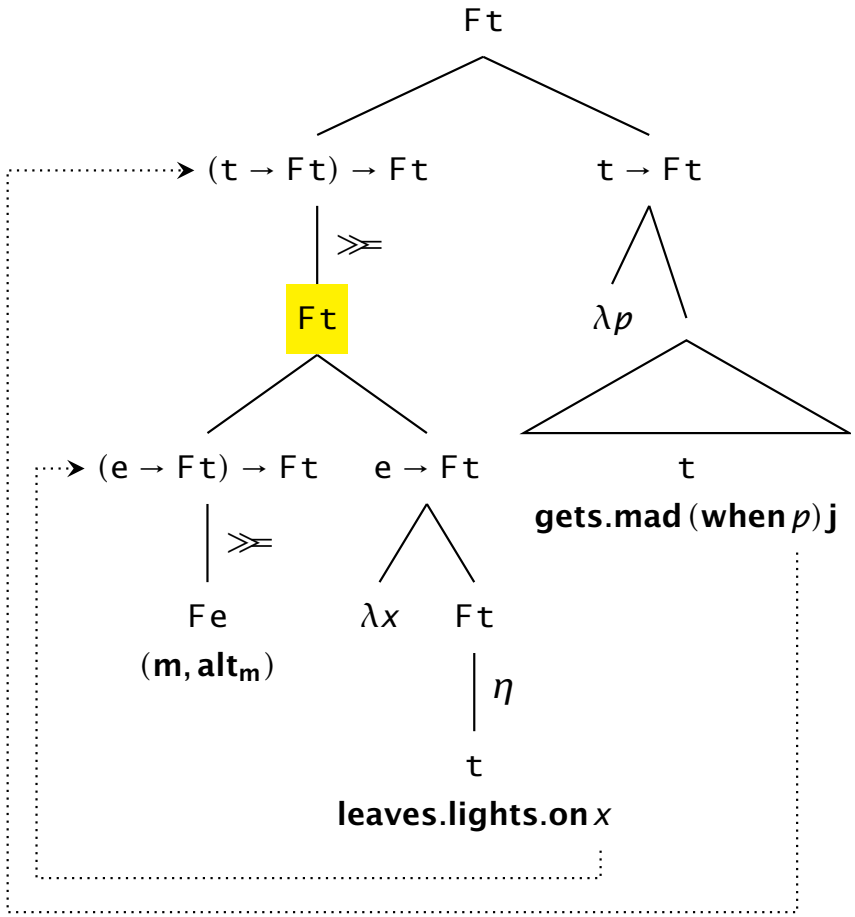
Similarly, focus interpretation can be construed categorematically:

$$\cdot \sim C :: \mathsf{F}\, a \to a$$

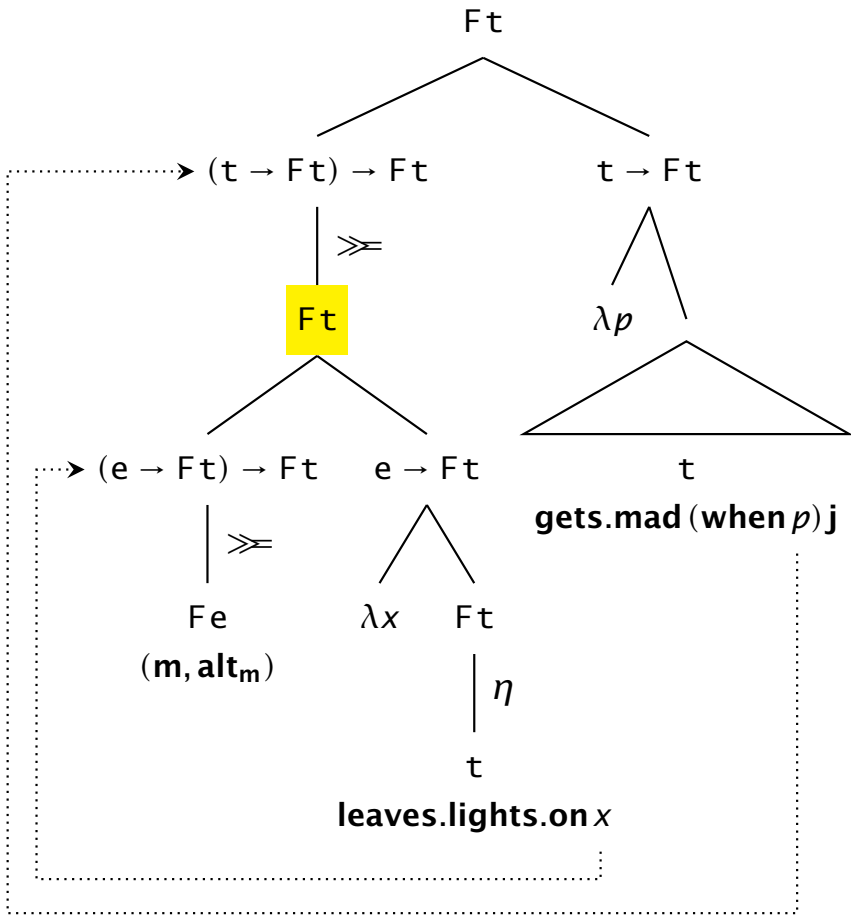
$$(x, A) \sim C := \begin{cases} x \text{ if } A \sim C \\ \# \text{ otherwise} \end{cases}$$

We're cheating a bit. Really, C represents a variable that's (dynamically) valued in a context (cf. Rooth 1992, Charlow 2015). So the semantics of focus interpretation really requires some apparatus for binding.

Islands


Ft
(t → Ft) → Ft
t → Ft
≫=
Ft
λp
(e → Ft) → Ft
e → Ft
t
gets.mad (when p) j
≫=
Fe
(m, alt_m)
λx
Ft
η
t
leaves.lights.on x

Islands



$$(\mathbf{mad}\,(\mathbf{when}\,(\mathbf{llo}\,\mathbf{m}))\,\mathbf{j},\ \{\mathbf{mad}\,(\mathbf{when}\,(\mathbf{llo}\,x))\,\mathbf{j} \mid x \in \mathbf{alt_m}\})$$

Selectivity

As ever, higher-order meanings for focused expressions allow us to secure selectivity outside scope islands:

$$m \gg\!= \lambda x.\, \eta\, (n \gg\!= \lambda y.\, \eta\, (\ldots x \ldots y \ldots)) :: \mathsf{F}\,(\mathsf{F}\,a)$$

An example of a higher-order focus meaning:

$$\Big((\mathbf{saw\,m\,j}, \{\mathbf{saw}\,x\,\mathbf{j} \mid x \in \mathbf{alt_m}\}), \Big\{ (\mathbf{saw\,m}\,y, \{\mathbf{saw}\,x\,y \mid x \in \mathbf{alt_m}\}) \mid y \in \mathbf{alt_j} \Big\} \Big)$$

Selectivity

As ever, higher-order meanings for focused expressions allow us to secure selectivity outside scope islands:

$$m \gg\!= \lambda x.\, \eta\, (n \gg\!= \lambda y.\, \eta\, (\ldots x \ldots y \ldots)) :: \mathsf{F}\,(\mathsf{F}\,a)$$

An example of a higher-order focus meaning:

$$\Big((\mathbf{saw\,m\,j}, \{\mathbf{saw}\,x\,\mathbf{j} \mid x \in \mathbf{alt_m}\}), \Big\{ (\mathbf{saw\,m}\,y, \{\mathbf{saw}\,x\,y \mid x \in \mathbf{alt_m}\}) \mid y \in \mathbf{alt_j} \Big\} \Big)$$

OMG.

Interaction

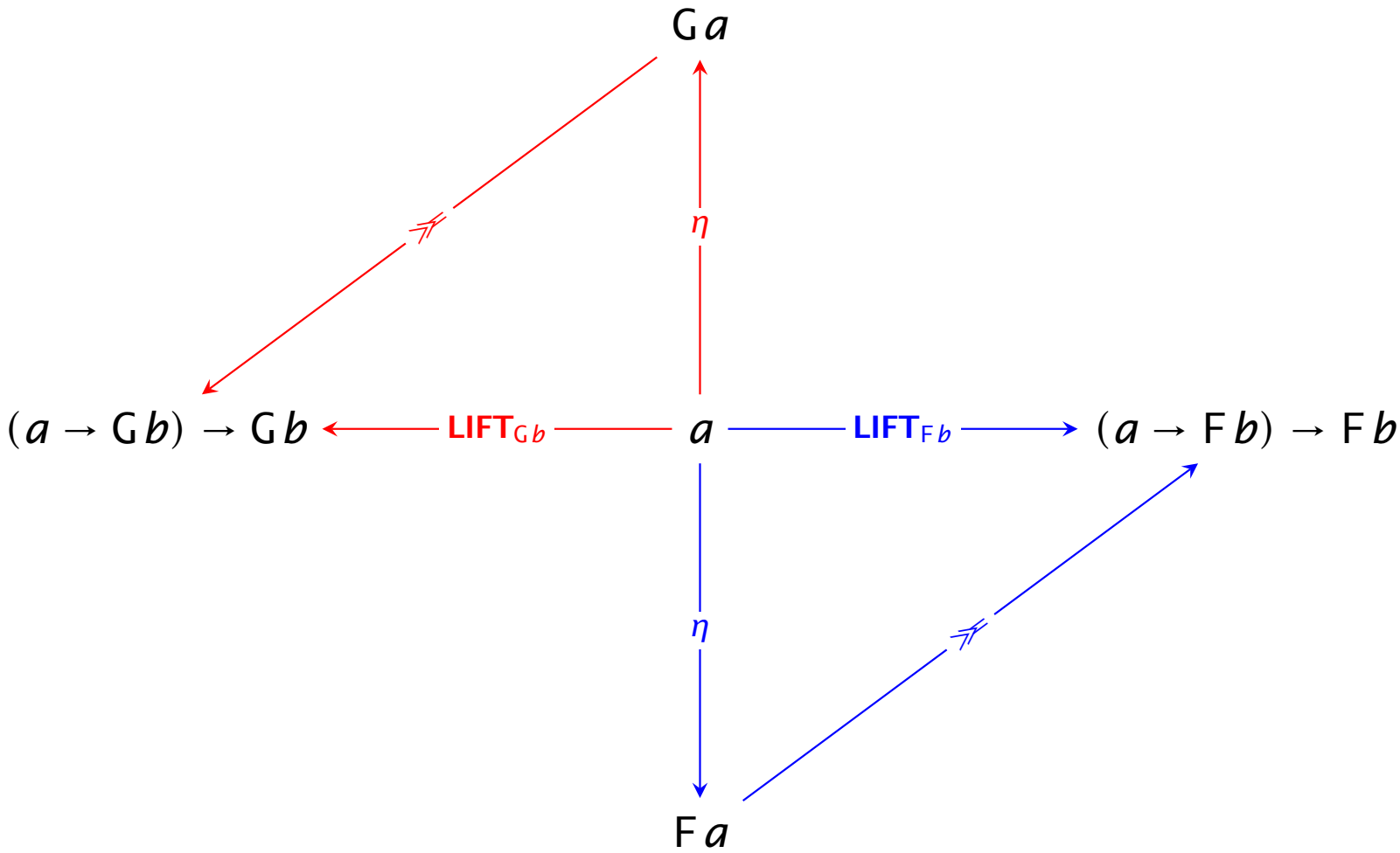
Binding

We've encountered a number of possible treatments of binding:

- Static: $\mathsf{G}\,a ::= \mathsf{g} \to a$ (the **Reader** monad)
- Dynamic: $\mathsf{H}\,a ::= \mathsf{g} \to (a, \mathsf{g})$ (the **State** monad)
- Static + alternatives: $\mathsf{GS}\,a ::= \mathsf{g} \to \mathsf{S}\,a$ (the **Reader-Set** monad)
- Dynamic + alternatives: $\mathsf{D}\,a ::= \mathsf{g} \to \mathsf{S}\,(a, \mathsf{g})$ (the **State-Set** monad)

Let's consider how the Focus monad might interact with these. We'll confine our attention to the simplest monad, G.

How they [don't] relate...



Two possibilities for layering

As with supplements, we can layer these monads on top of each other, without any need to directly define a combined monad:

Assignment-dependent focused meanings: $\mathsf{G}\,(\mathsf{F}\,a)$
Focused assignment-dependent meanings: $\mathsf{F}\,(\mathsf{G}\,a)$

Generating both of these orderings:

$$m \gg\!= \lambda x.\, \eta\,(n \gg\!= \lambda y.\, \eta\,(\ldots\, x\, \ldots\, y\, \ldots)) :: \mathsf{G}\,(\mathsf{F}\,a)$$
$$m \gg\!= \lambda x.\, \eta\,(n \gg\!= \lambda y.\, \eta\,(\ldots\, x\, \ldots\, y\, \ldots)) :: \mathsf{F}\,(\mathsf{G}\,a)$$

Assignments over alternatives

Here's a sample meaning of type $\mathsf{G}(\mathsf{F}\,\mathtt{t})$:

$$\lambda g.\,(\mathbf{saw}\,g_0\,\mathbf{m},\,\{\mathbf{saw}\,g_0\,x \mid x \in \mathbf{alt_m}\})$$

Here's another possibility, type $\mathsf{G}(\mathsf{F}\,\mathtt{e})$ for a *focused pronoun*:

$$\lambda g.\,(g_0,\,\mathbf{alt}_{g_0})$$

Alternatives over assignments

Here's a sample meaning of type $\mathsf{F}(\mathsf{G}\,\mathtt{t})$:

$$(\lambda g.\,\mathbf{saw}\,g_0\,\mathbf{m},\ \{\lambda g.\,\mathbf{saw}\,g_0\,x \mid x \in \mathbf{alt_m}\})$$

Here's another possibility, type $\mathsf{F}(\mathsf{G}\,\mathtt{e})$ for a *focused pronoun*:

$$\left(\lambda g.\,g_0,\ \mathbf{alt}_{\lambda g.\,g_0}\right)$$

Visualizing higher-order meanings

We can represent higher-order meanings (or monadic meanings more generally) using a handy "tower" notation. For example, our two meanings for focused pronouns can be represented as follows:

$$\begin{array}{c} \lambda g.[\ ] \\ \hline ([\ ], \mathbf{alt}_{g_0}) \\ \hline g_0 \\ \mathsf{G}\,(\mathsf{F}\,\mathsf{e}) \end{array} \qquad \begin{array}{c} ([\ ], \mathbf{alt}_{\lambda g.\, g_0}) \\ \hline \lambda g.[\ ] \\ \hline g_0 \\ \mathsf{F}\,(\mathsf{G}\,\mathsf{e}) \end{array}$$

When you get down to it, both of these have g_0 as their "core" value. But the layerings of the effects are different, as are the invoked alternatives.

For flexibility?

Having two separate monads, one for focus, and one for pronouns, allows us to generate both layerings of focus and pronominal effects: $\mathsf{G}(\mathsf{F}\,a)$, and $\mathsf{F}(\mathsf{G}\,a)$. Is there evidence that such flexibility is warranted?

Arguments from Jacobson (2000, 2004) suggest it might be:

1. Every third grade boy loves Mary.
   And every FOURTH$_{\mathsf{F}}$ grade boy loves himSELF$_{\mathsf{F}}$.
2. Every third grade boy loves himself.
   And every FOURTH$_{\mathsf{F}}$ grade boy loves HIMself$_{\mathsf{F}}$.

Jacobson suggests that the *himSELF* accent pattern is associated with (in our terms) $\mathsf{G}(\mathsf{F}\,a)$, and the *HIMself* accent pattern with $\mathsf{F}(\mathsf{G}\,a)$.

Summing up

So, at least with focus and pronouns, there is little reason to assume a combined monad. It is enough to toss both monads into your grammar and let the chips fall where they may.

The two-monads approach predicts two substantively different layerings of focus and pronominal effects, and we have some initial evidence (stress patterns on bound reflexive pronouns) that this may be correct.

Obviously, this is all very preliminary, pending further investigation. . .

Functors (and structured meanings)

Monads, etc

We have seen several constructs for characterizing how "enriched" meanings interact with others in semantic composition.

- **Monads** allow us to sequence an enriched thing with a scope:

$$\gg\!= :: T\,a \to (a \to T\,b) \to T\,b$$

- **Applicatives** give rise to a notion of enriched functional application:

$$\backslash\backslash :: T\,a \to T\,(a \to b) \to T\,b$$

- **Functors** allow us to map a function over some structure:

$$\bullet :: T\,a \to (a \to b) \to T\,b$$

Relative strength

Every monad is applicative, and every applicative is a functor:

Monad $\subset$ **Applicative** $\subset$ **Functor**

Against monadic (and applicative) treatments of focus

A sentence like *JOHN$_F$ saw MARY$_F$* is only felicitous as a response to a question like *who saw whom?* or a sentence like *Bill saw Sue*.

If you try using *JOHN$_F$ saw MARY$_F$* to reply to *who saw Mary?* or *Bill saw Mary*, over-focusing results in infelicity.

It is difficult to see how this can be ruled out with a type $\mathsf{F}\,\mathsf{t}$ meaning:

$$(\mathbf{saw}\,\mathbf{m}\,\mathbf{b}, \{\mathbf{saw}\,y\,x \mid x \in \mathbf{alt_b}, y \in \mathbf{alt_m}\})$$

But this is precisely the kind of meaning generated in monadic (as well as applicative) approaches to F!

Higher-order meanings to the rescue?

Higher-order meanings are finer-grained than "flat" focus structures:

$$\Big((\mathbf{saw}\,\mathbf{m}\,\mathbf{j}, \{\mathbf{saw}\,x\,\mathbf{j} \mid x \in \mathbf{alt_m}\}), \Big\{(\mathbf{saw}\,\mathbf{m}\,y, \{\mathbf{saw}\,x\,y \mid x \in \mathbf{alt_m}\}) \mid y \in \mathbf{alt_j}\Big\}\Big)$$

This extra resolution may allow us to rule out over-focusing (if interested, you can see a sketchy implementation here).

But for this to work, higher-order meanings have to be the *only* option (since flat meanings illicitly allow over-focusing).

Making higher-order meanings a default

If $\gg=$ is replaced with F's mapping operation $\bullet$, higher-order meanings with multiply-focused constructions will be the only option:

$$(x, X) \bullet f := (f\,x, \{f\,x' \mid x' \in X\})$$

A schematic example:

$$m \bullet \lambda x.\, n \bullet \lambda y. \ldots\, x \ldots\, y \ldots :: \mathrm{F}\,(\mathrm{F}\,a)$$

Still predicts island insensitivty (and, naturally, selectivity):

$$(m \bullet \lambda x.\, f\,x) \bullet g = m \bullet (\lambda x.\, g\,(f\,x))$$

Functors for structured meanings

As long as we're using functors, it's interesting to note that structured meaning composition can be characterized in terms of functors:

$$(x, f) \bullet g := (x, g \circ f)$$

Then the meaning of F-marks can be given as follows:

$$\cdot_{\mathbf{F}} := \lambda x.\,(x, \lambda y.\,y)$$

A sample derivation for *JOHN*$_F$ *saw MARY*$_F$:

$$\mathbf{j_F} \bullet \lambda x.\,\mathbf{m_F} \bullet \lambda y.\,\mathbf{saw}\,y\,x = (\mathbf{j}, \lambda x.\,(\mathbf{m}, \lambda y.\,\mathbf{saw}\,y\,x))$$

References I

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

Charlow, Simon. 2015. Givenness, compositionally and dynamically. In Eric Baković (ed.), *Short 'schrift for Alan Prince*. http://princeshortschrift.wordpress.com/squibs/charlow.

Jacobson, Pauline. 2000. Paychecks, stress, and variable-free semantics. In Brendan Jackson & Tanya Matthews (eds.), *Proceedings of Semantics and Linguistic Theory 10*, 65–82. Ithaca, NY: Cornell University. http://dx.doi.org/10.3765/salt.v10i0.3103.

Jacobson, Pauline. 2004. Kennedy's puzzle: What I'm named or who I am?. In Kazuha Watanabe & Robert B. Young (eds.), *Proceedings of Semantics and Linguistic Theory 14*, 145–162. Ithaca, NY: Cornell University. http://dx.doi.org/10.3765/salt.v14i0.2911.

Krifka, Manfred. 1991. A compositional semantics for multiple focus constructions. In Steve Moore & Adam Wyner (eds.), *Proceedings of Semantics and Linguistic Theory 1*, 127–158. Ithaca, NY: Cornell University.

Krifka, Manfred. 2006. Association with focus phrases. In Valéria Molnár & Susanne Winkler (eds.), *The Architecture of Focus*, 105–136. Mouton de Gruyter.

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

References II

Rooth, Mats. 1992. Ellipsis redundancy and reduction redundancy. In Steven Berman & Arild Hestvik (eds.), *Proceedings of the Stuttgart Workshop on Ellipsis, no. 29 in Arbeitspapiere des SFB 340*. Stuttgart: University of Stuttgart.

Shan, Chung-chieh. 2002. Monads for natural language semantics. In Kristina Striegnitz (ed.), *Proceedings of the ESSLLI 2001 Student Session*, 285–298. http://arxiv.org/abs/cs/0205026.