Continuation semantics ii[1]

Patrick D. Elliott[2] *& Martin Hackl*[3]

March 12, 2020

[1] 24.979: Topics in semantics

Getting high: Scope, projection, and evaluation order

[2] `pdell@mit.edu`

[3] `hackl@mit.edu`

Homework

- Finish reading **barkerShan2015** chapters 1 and 4, if you haven't already.
- Read **barkerShan2015** chapter 7.
- Do p-set 2 (to be posted on Stellar later today)!

1 *Roadmap*

This week, I'll finish introducing *continuation semantics*. We'll minimally try to cover:

- The syntax-semantics interface (more explicitly, this time).
- Inverse scope via multi-story towers.
- Split scope.
- Scope islands as *evaluation islands* + remarks on scope economy.
- Generalized con/dis-junction + "split scope" readings.

With potentially two extensions:

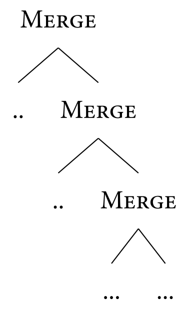
- DP-internal composition and indexed continuations.
- Exceptionally-scoping indefinites via continuations (**Charlowc**).

2 *A note on syntax*

So far, I've been a little shy about saying explicitly what we're assuming here about syntax, and what we're assuming about the syntax-semantics mapping.

I'll assume a derivational theory, according to which structures are built-up via successive application of Merge.[4]

[4] I'll often use "syntactic structure speak" when talking about trees. This is harmless, since they can always be interpreted as the graph of a syntactic derivation, especially since trees encode both structure and order.

(1)


I'll furthermore adopt the hypothesis that the syntactic derivation proceeds in lockstep with the semantic computation. This conjecture, which goes back at least to **montague1973**, is often described as *direct compositionality*.[5]

[5] Although direct compositionality is often associated with frameworks such as variable free semantics and Combinatory Categorial Grammar, it's in principle independent. See, e.g., **kobele2006** for an explicit formalization of a directly compositional minimalist grammar.

Minimally, the formatives must be *tuples* consisting of phonological features and semantic features: $(\mathtt{phon}, \mathtt{sem}, ...)$. Semantic features could be cashed out as model theoretic objects, or perhaps as expressions of the simply typed lambda calculus.

I'll assume that part of what Merge does is concatenate phonological features. This is because Merge is just an instruction for combining formatives. On the semantic side, it typically does function application.[6]

[6] It follows that Merge is a *non-symmetric relation*, departing from, e.g., **chomsky1995**, but consistent with **stabler1997** and related work.

(2) $(\mathbb{x}, x) \; * \; (\mathbb{y}, y) := ([\mathbb{x}\ \mathbb{y}], x \mathsf{A}\, y)$

It can also do concatenation of phonological features, plus **sfa!** (**sfa!**) of semantic values (whence the left-to-right bias of S).

(3) $(\mathbb{x}, x) * (\mathbb{y}, y) := ([\mathbb{x}\ \mathbb{y}], x \mathsf{S}\, y)$

I've define LIFT as a purely semantic operation – this is to be taken as shorthand for an operation on a formative that only effects the semantic value:

(4) $(\mathbb{x}, x)^{\uparrow} := (\mathbb{x}, x^{\uparrow})$

This constitutes the basics of the system laid out in **elliott2019movement**. See **elliott2019movement** for an elaboration of how to supplement this system with a feature calculus, but for today's purposes we won't need any additional assumptions.

2.1 Deriving inverse scope

Last type we got as far as being able to derive *surface scope* readings, as well as scopal ambiguities arising via interactions with scopally-immobile expressions.

We achieved this latter coverage by allowing LOWER to fix the scope of a quantifier at different points in the derivation.

We still don't have any way of accounting for scopal interactions between multiple scopally-mobile expressions (i.e., quantifiers). Since every major theory of quantifier scope is tailored to achieve this, we have a major problem on our hands!

Fortunately, we already have all of the primitive operations we need in order to achieve inverse scope readings.

Recall that LIFT is a *polymorphic function* – it lifts a value into a trivial tower:

(5) $a^{\uparrow} := \dfrac{[]}{a}$

Since LIFT is polymorphic, in principle it can apply to any kind of value – even a tower! Let's flip back to lambda notation to see what happens.

(6) $[\![\text{everyone}]\!] := \lambda k\,.\,\forall x[k\ x]$ $\qquad \texttt{(e} \to \texttt{t)} \to \texttt{t}$

(7) $[\![\text{everyone}]\!]^{\uparrow} = \lambda l\,.\,l\,(\lambda k\,.\,\forall x[k\ x])$ $\qquad \texttt{(((e} \to \texttt{t)} \to \texttt{t)} \to \texttt{t)} \to \texttt{t}$

Going back to tower notation, lifting a tower adds a trivial third story:[7] Following **Charlowc**, when we apply LIFT to a tower, we'll describe the operation as *external lift* (although, it's worth bearing in mind that this is really just our original LIFT function).

[7] In fact, via successive application of LIFT, we can generate an $n-$story tower.

The third story of the tower corresponds to whatever takes scope over the outer continuation variable (here, l), and the second story of the tower corresponds to whatever takes scope over the inner continuation variable (here, k).

(8) $\left(\dfrac{\forall x[]}{x}\right)^{\uparrow} = \dfrac{[]}{\dfrac{\forall x[]}{x}}$

One important thing to note is that, when we externally lift a tower, the quantificational part of the meaning always remains on the same story relative to the bottom story. Intuitively, this reflects the fact that, ultimately, LIFT alone isn't going to be enough to derive quantifier scope ambiguities.

Question

Which (if any) of the following bracketings make sense for a three-story tower:

(9) $$\frac{\left(\dfrac{f\,[]}{g\,[]}\right)}{x}$$

(10) $$\frac{f\,[]}{\left(\dfrac{g\,[]}{x}\right)}$$

The extra ingredient we're going to need in order to derive inverse scope, is the ability to sandwhich an empty story into the *middle* of our tower, pushing the quantificational part of the meaning to the very top.

This is *internal lift* ($\upuparrows$).[8]

[8] I can tell what you're thinking: "seriously? Another *darn* type-shifter? How many of these are we going to need?!". Don't worry, I got you. Even thought we've defined internal lift here as a primitive operation, it actually just follows from our existing machinery. Concretely, *internal lift* is really just *lifted* LIFT (so many lifts!). Lifted LIFT applies to its argument via S.

(11) $$\frac{[]}{\uparrow}\,\mathsf{S}\,\frac{f\,[]}{x} = \frac{f\,[]}{\dfrac{[]}{x}}$$

(12) *Internal lift* (def.)
  a. $(\upuparrows) : \mathsf{K_t\ a \to K_t\ (K_t\ a)}$
  b. $m^{\upuparrows} := \lambda k\ .\ m\ (\lambda x\ .\ k\ x^{\uparrow})$

It's much easier to see what internal lift is doing by using the tower notation. We can also handily compare its effects to those of *external* lift.

(13) *Internal lift* (tower ver.)

$$\left(\frac{f\,[]}{x}\right)^{\upuparrows} := \frac{f\,[]}{\dfrac{[]}{x}}$$

(14) *External lift* (tower ver.)

$$\left(\frac{f\,[]}{x}\right)^{\uparrow} := \frac{[]}{\dfrac{f\,[]}{x}}$$

Armed with *internal* and *external* lifting operations, we now have everything we need to derive inverse scope. We'll start with a simple example (**??**).

The trick is: we *internally* lift the quantifier that is destined to take wide scope.

(15) A boy danced with every girl. $\forall > \exists$

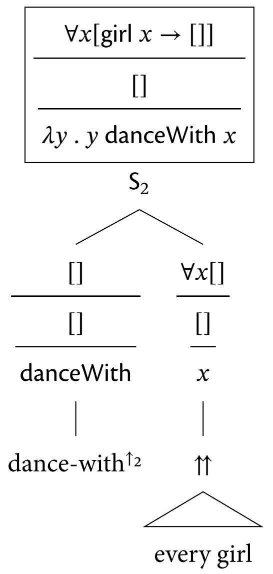
Before we proceed, we need to generalize LIFT and **SFA!** to three-story towers.[9]

[9] Before you get worries about expanding our set of primitive operations, notice that *3-story lift* is just ordinary lift applied twice. *3-story* **SFA!** is just **SFA!**, but where the bottom story combines via S not A. In fact, we can generalize these operations to $n-$story towers.

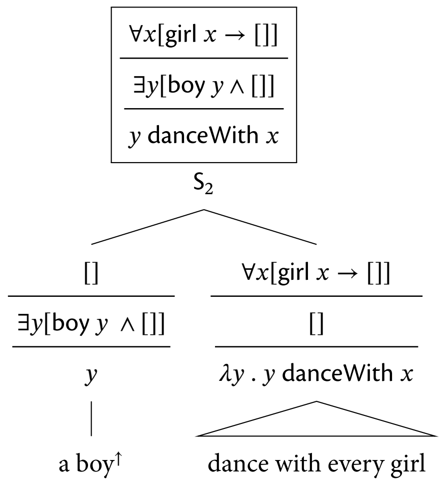
(16) $x^{\uparrow_2} := \begin{array}{c} [] \\ \hline [] \\ \hline x \end{array}$

(17) $\begin{array}{c} f\,[] \\ \hline m \end{array} \; \mathsf{S}_2 \; \begin{array}{c} g\,[] \\ \hline n \end{array} := \begin{array}{c} f\,(g\,[]) \\ \hline m \; \mathsf{S} \; n \end{array}$

(18) Step 1: internally lift *every girl*



(19) Step 2: *ex*ternally lift *a boy*



What we're left with now is a 3-story tower with the universal on the top story and the existential on the middle story. We can collapse the tower by first collapsing the bottom two stories, and then collapsing the result. In order to do this, we'll first define *internal lower*.[10]

[10] Let's again address the issue of expanding our set of primitive operations (in what is becoming something of a theme). Internal lower is just lifted lower, applying via S. In other words:

(20) $m^{\Downarrow} \equiv (\downarrow)^{\uparrow} \; \mathsf{S} \; m$

(21) *Internal lower* (def)

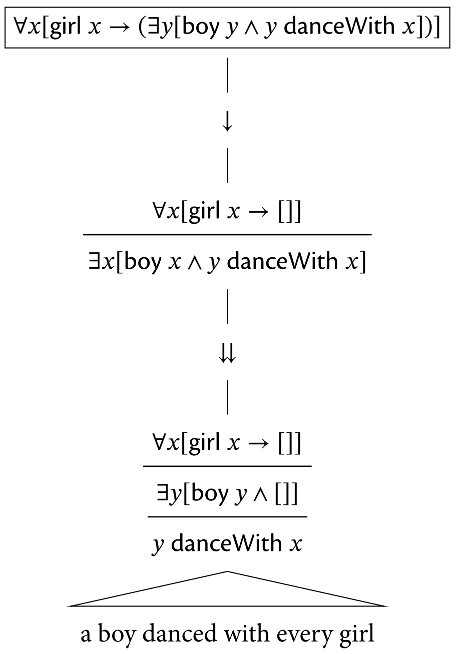
a. $(\Downarrow) : \mathsf{K_t}\;(\mathsf{K_t}\;\mathsf{a}) \rightarrow \mathsf{K_t}\;\mathsf{a}$

b. $m^{\Downarrow} := \lambda k \,.\, m\,(\lambda n \,.\, k\, n^{\downarrow})$

(22) *Internal lower* (def.) $\left(\begin{array}{c} f\,[] \\ \hline g\,[] \\ \hline p \end{array}\right)^{\Downarrow} := \begin{array}{c} f\,[] \\ \hline \left(\begin{array}{c} g\,[] \\ \hline p \end{array}\right)^{\downarrow} \end{array}$

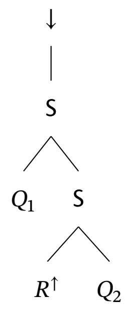
Now we can collapse the tower by doing *internal lower*, followed by *lower*:

(23)

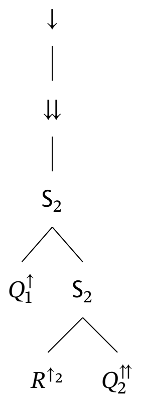

$\boxed{\forall x[\text{girl } x \rightarrow (\exists y[\text{boy } y \wedge y \text{ danceWith } x])]}$
$\downarrow$
$\forall x[\text{girl } x \rightarrow []]$
$\exists x[\text{boy } x \wedge y \text{ danceWith } x]$
$\Downarrow$
$\forall x[\text{girl } x \rightarrow []]$
$\exists y[\text{boy } y \wedge []]$
$y \text{ danceWith } x$
a boy danced with every girl

Great! We've shown how to achieve quantifier scope ambiguities using our new framework. Let's look at the derivations again side-by-side.

(24) Surface scope (schematic derivation)



(25) Inverse scope (schematic derivation)



There's a couple of interesting things to note here:

- The inverse scope derivation involves more applications of our type-shifting operations – this becomes especially clear if we decompose the complex operations S_2, $\uparrow_2$, $\uparrow\uparrow$, and $\Downarrow$.
- In order to derive an inverse scope reading, what was *crucial* was the availability of *internal lift*; the remaining operations, S_2, $\uparrow_2$, $\Downarrow$ only functioned to

massage composition for three-story towers.

On the latter point, it's tempting to conjecture that in, e.g., German, Japanese and other languages which "wear their LF on their sleeve", the semantic correlate of *scrambling* is *internal lift*, whereas in scope-flexible languages such as English, internal lift is a freely available operation.[11]

[11] To make sense of this, we would of course need to say something more concrete about overt movement. For an attempt at marrying continuations to a standard, minimalist syntactic component, see my manuscript *Movement as higher-order structure building*.

If we adopt some version of the *derivational complexity hypothesis*[12], we also predict that inverse scope readings should take longer to process than surface scope readings.

[12] I.e., that derivational complexity correlates with processing difficulty.

It's worth mentioning, incidentally, that although we collapsed the resulting three-story tower via internal lower followed by lower, we can also define an operation that collapses a three-story tower two an ordinary tower in a different way. Let's call it *join*:[13]

[13] Join for three-story towers corresponds directly to the *join* function associated with the continuation monad. For more on continuations from a categorical perspective, see the appendix to the first handout.

(26) *join* (def.)
$m^{\mu} \coloneqq \lambda k \,.\, m\,(\lambda c \,.\, c\; k)$ $\qquad \mu : \mathsf{K_t}\;(\mathsf{K_t}\;\mathsf{a}) \to \mathsf{K_t}\;\mathsf{a}$

In tower terms, join takes a three-story tower and sequences quantifiers from top to bottom:

(27) $$\left(\frac{f\,[]}{\dfrac{g\,[]}{x}}\right)^{\mu} = \frac{f\,(g\,[])}{x}$$

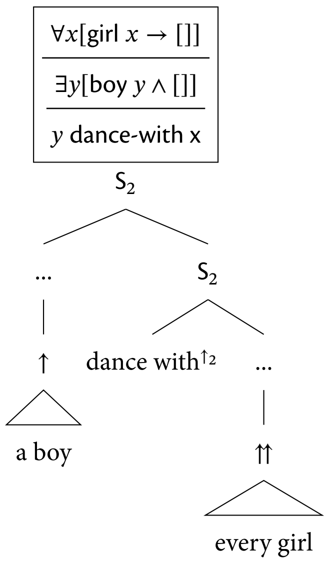
Doing internal lower on a three-story tower followed by lower is equivalent to doing join on a three-story tower followed by lower (as an exercise, convince yourself of this). However, there's may be a good empirical reason for having internal lower as a distinct operation (and since it's just lifted lower, it "comes for free" in a certain sense).

(28) Daniele wants a boy to dance with every girl. $\qquad \forall > \mathsf{want} > \exists$

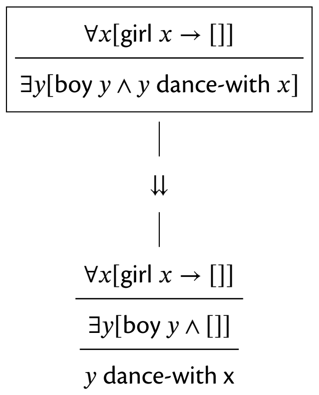
Arguably, (**??**) can be true if for every girl x, Daniele has the following desire: *a boy dances with y*. This is the reading on which *every boy* scopes over the intensional verb, and *a boy* scopes below it.

If we have *internal lower*, getting this is easy. We *internally lift every girl* and *externally lift a boy*. Before we reach the intensional verb, we fix the scope of *a* *boy* by doing internal lower. Now we have an ordinary tower, and we can defer fixing the scope of *every girl* via *lower* until after the intensional verb.

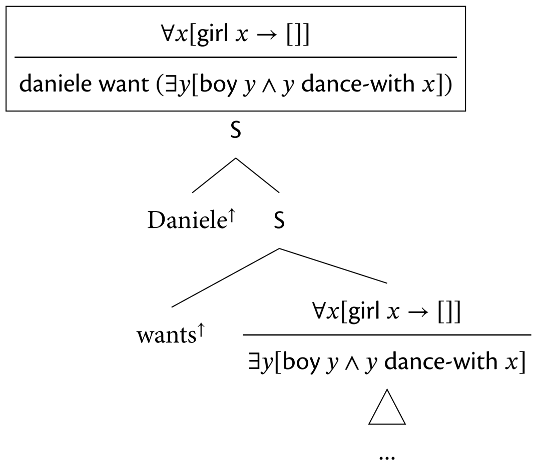
(29) Step 1: scope *every girl* over *a boy*



(30) Step 2: internally lower below *want*



(31) Step 3: scope *every girl* over *want*


∀x[girl x → []]
daniele want (∃y[boy y ∧ y dance-with x])
S
Daniele↑
S
wants↑
∀x[girl x → []]
∃y[boy y ∧ y dance-with x]
...

(32) Step 4: lower

$$\boxed{\forall x\begin{bmatrix}\textsf{girl}\ x \\ \to \left(\textsf{daniele want}\left(\exists y\begin{bmatrix}\textsf{boy}\ y \\ \wedge\ y\ \textsf{dance-with}\ x\end{bmatrix}\right)\right)\end{bmatrix}}$$

$$\Big\downarrow$$

$$\frac{\forall x[\textsf{girl}\ x \to [\,]]}{\textsf{daniele want}\ (\exists y[\textsf{boy}\ y \wedge y\ \textsf{dance-with}\ x])}$$

If we only have *join* then the scope of *a boy* and *every girl* may vary amongst themselves, but they should either both scope below *want* or both scope above *want*.

Note that we haven't given a concrete treatment of intensionality here, but, mechanically, we can simply replace every occurrence of $\texttt{t}$ with type $\texttt{s} \to \texttt{t}$. Intensional verbs take type $\texttt{s} \to \texttt{t}$ complements. Here are some sample lexical entries:

(33) a. $[\![\text{every girl}]\!] := \lambda k\,.\,\lambda w\,.\,\forall x[\textsf{girl}_w\ x \to k\ x]$ $\quad (\texttt{e} \to (\texttt{s} \to \texttt{t})) \to \texttt{s} \to \texttt{t}$
b. $[\![\text{dance with}]\!] := \lambda yx\,.\,\lambda w\,.\,y\ \textsf{dance-with}_w\ x$ $\quad \texttt{e} \to \texttt{s} \to \texttt{t}$
c. $[\![\text{want}]\!] := \lambda p\,.\,\lambda x\,.\,\lambda w\,.\,x\ \textsf{want}_w\ p$ $\quad (\texttt{s} \to \texttt{t}) \to \texttt{e} \to \texttt{s} \to \texttt{t}$

Instead of giving back a return type $\texttt{t}$, in an intensional setting our continuation type constructor is defined in terms of a return type $\texttt{s} \to \texttt{t}$:

(34) $\texttt{K}_{\texttt{s}\to\texttt{t}}\texttt{a} := (\texttt{a} \to (\texttt{s} \to \texttt{t})) \to \texttt{s} \to \texttt{t}$

You can verify for yourselves that the core features of the system remain unaffected, i.e, the definitions of lift and S can remain the same.

3 *Split scope*

In the first p-set, I asked you how to think about analyzing split scope of non-upward-monotone quantifiers:

(35) The company need fire no employees.

It's not the case that the company needs to hire employees. $\neg > \square > \exists$

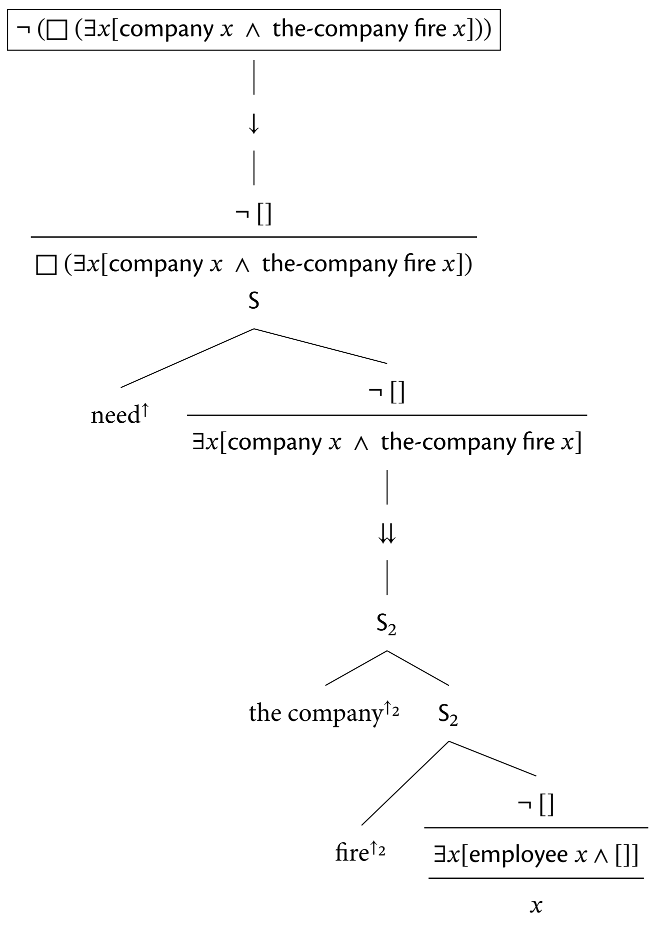
With continuation semantics, we can understand this data as providing support for the idea that expressions can denote three-story towers (something not excluded by, e.g., **heimKratzer1998** in any case).

(36) $[\![\text{no employees}]\!] := \lambda k\,.\,\neg\, k\,(\lambda l\,.\,\exists x[\text{employee } x \wedge l\ x])$ $\quad \mathtt{K_t\ (K_t\ a)}$

Tower version:

(37) $$[\![\text{no employees}]\!] := \begin{array}{c} \neg\,[] \\ \hline \exists x[\text{employee } x \wedge []] \\ \hline x \end{array}$$

We get the split scope reading by doing internal lower first below the modal, and then external lower above the modal.

(38)

¬ (□ (∃x[company x ∧ the-company fire x]))
↓
¬ []
□ (∃x[company x ∧ the-company fire x])
S
need↑
¬ []
∃x[company x ∧ the-company fire x]
⇊
S₂
the company↑₂
S₂
fire↑₂
¬ []
∃x[employee x ∧ []]
x

One interesting property of split scope readings is that it is essential that we be able to *lower* a 3-story tower in two distinct steps – *internal lower* followed by *lower*.

Despite being very elegant, I'm not sure whether this is a completely satisfactory account of split scope readings – it's been observed, for example, that split scope always seems to involve narrow scope of an existential (see, e.g., **abelsMarti2010**). This doesn't fall out from the analysis outlined here.[14]

[14] An investigation of split scope from the perspective of continuations could be a good project for this class – see, e.g., **bumford2017** for relevant discussion.

4 *Scope islands and obligatory evaluation*

Quantifiers can't take scope arbitrarily high – rather, their scope is roofed by certain constituents. As is well known, the environments that are *islands for scope taking* don't necessarily correspond to the environments that are islands for overt movement (see **may**).

We still need a theory of scope islands in order to restrict the power of continuation semantics. It turns out that there is a very natural notion available to us, similar to **chomsky1995**'s notion of a *phase*.

Inspired by research on *delimited control* in computer science[15], **Charlowc** develops an interesting take on scope islands couched in terms of continuations.

[15] See, e.g., **danvyFilinski1992** and **wadler1994**.

He proposes the following definition:

(39) *Scope islands* (def.)
A *scope island* is a constituent that is subject to *obligatory evaluation*.
(**Charlowc**)

By *obligatory evaluation*, we mean that every continuation argument *must* be saturated before semantic computation can proceed. In other words, a scope island is a constituent where, if we have something of type $\texttt{K}_\texttt{t}\ \texttt{a}$, we must lower it before we can proceed.

Let's be more precise:

(40) a. A constituent X is *evaluated* if it has an evaluated type $\texttt{a}$.
b. A type $\texttt{a}$ is *evaluated* if $\texttt{a} \neq \texttt{K}_\texttt{t}\ \texttt{b}$.

One way of thinking about this, is that the presence of an unsaturated continuation argument means that there is some computation that is being deferred until later.

Scope islands are constituents at which evaluation is *forced*. As noted by **Charlowc**, this idea bears an intriguing similarity to **chomskyPhase**'s notion of a *phase*.[16]

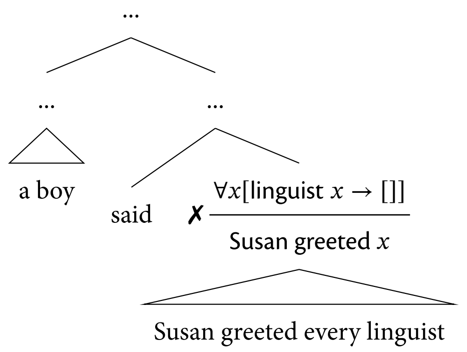
[16] Exploring this parallel in greater depth could make for an interesting term paper topic.

How does this work in practice? A great deal of ink has been spilled arguing that, e.g., a finite clause is a scope island.

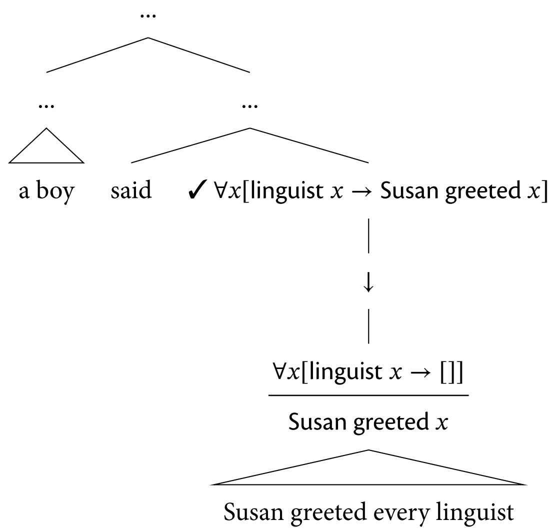
(41) A boy said $\overbrace{\boxed{\text{that Susan greeted every linguist}}}^{\text{scope island}}$. $\exists > \forall$; ✗ $\forall > \exists$

The derivation of the embedded clause proceeds as usual via lift and **sfa!**.

(42) Scope island with an unevaluated type



(43) Scope island with an evaluated type


...
...
...
a boy
said
✓ ∀x[linguist x → Susan greeted x]
↓
∀x[linguist x → []]
Susan greeted x
Susan greeted every linguist

This story leaves a lot of questions unanswered of course:

- Is this just a recapitulation of a representational constraint on quantifier raising?[17]
- Are finite CPs the only scope island? What about DPs?[18]
- Can we give a principled story about islands for overt movement using similar mechanisms? What explains the difference between overt movement and scope taking with respect to locality?[19]

[17] The answer to this question may ultimate be *yes*, in my view.

[18] In my view, yes, the default assumption should be that DPs are scope islands. See **sauerland2005dp** and **charlow2010** for relevant discussion.

[19] If we want to give a more general account of phases using this mechanism, we need to give an account of overt movement in terms of continuations, too. See my unpublished ms. *Movement as higher-order structure building* for progress in this direction.

4.1 A brief remark on Scope Economy

As famously discovered by **fox1995**, scope-shifting operations are subject to an economy condition.

(44) Economy condition on scope shifting (*Scope Economy*) (def.)
OP can apply only if it affects semantic interpretation (i.e., only if inverse-scope and surface-scope are semantically distinct).[20]
(**fox2000**)

[20] Here, OP stands for *scope-shifting operation*, i.e., quantifier raising.

Scope economy has an impressively wide empirical coverage, including interactions between scope and ellipsis. Consider, e.g., the following contrasts (from **fox1995**):

(45) Some linguist likes every philosopher. $\exists > \forall; \forall > \exists$

(46) Some linguist likes every philosopher,
and some mathematician does too. $\exists > \forall; \forall > \exists$

(47) Some linguist likes every philosopher,
and Mary does too. $\exists > \forall$; ✗ $\forall > \exists$

(48) Some linguist likes every philosopher,
and every mathematician does too. $\exists > \forall$; ✗ $\forall > \exists$

Scope economy gives us an explanation of this paradigm, just in case we assume that an elliptical sentence and its antecedent must involve *parallel scopal relations*.

In (**??**), for example, QR of *every philosopher* over *every mathematician* is blocked by scope economy, since the relative scope of two universal quantifiers doesn't affect truth-conditions.

(49) *every philosopher [λx every mathematician does $\boxed{\text{like } t_x}$]

Since the antecedent must involve a parallel scopal relation, inverse scope is blocked.

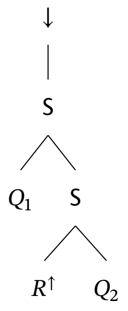
Here, I just want to point out that our continuation semantic framework is actually an extremely natural fit for scope economy:

(50) Economy condition on scope shifting (continuations ver.)

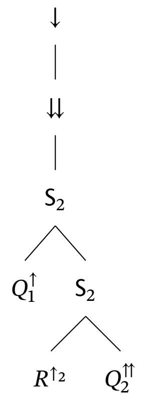
> A derivation D is ruled out if there is a simpler derivation D' that gives rise to the same interpretation.

Just so long as the economy condition is evaluated at scope islands (as argued for by **fox1995**), we capture the basic observations, since scope *every philosopher* over *every mathematician*, involves, minimally, an additional internal lift of *every philosopher*.

(51) Surface scope (schematic derivation)



(52) Inverse scope (schematic derivation)



It remains to be seen whether an economy condition framed in terms of the complexity of a continuation-semantic derivation makes any distinct predictions to an economy condition framed in terms of QR.[21]

[21] Figuring this out would be a great student project, I think.

5 *Generalized con/dis-junction*

The flexibility of *and* and *or* has been discussed at length by, e.g., **parteeRooth**, **winter_flexibility_2001**, among others.[22]

[22] We saw an example of this when we motivated LIFT.

(53) Lan and some woman arrived.

(54) Howie sneezed and/or coughed.

(55) Lan kissed and/or hugged Irene.

Unlike other expressions we've seen so far, we can characterize *and* and *or* as expressions that takes two continuized values as arguments.[23]

[23] Note that since the continuation argument k occurs more than once in the function body, we can no longer abbreviate the flat lambda-expression using a tower.

(56) a. m and $n \coloneqq \lambda k\,.\,m\,k\ \wedge n\,k$ and : $\mathsf{K_t\ a \to K_t\ a \to K_t\ a}$
b. m or $n \coloneqq \lambda k\,.\,m\,k\ \vee n\,k$ or : $\mathsf{K_t\ a \to K_t\ a \to K_t\ a}$

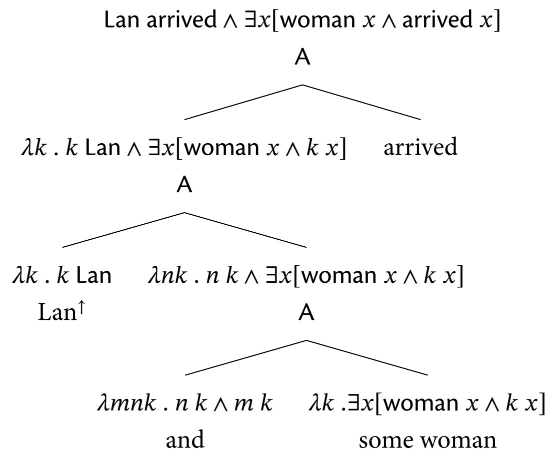
The intuition here is as follows: *and* wants as its arguments things that are *guaranteed* to give back truth values at some future stage of computation.

This accounts for the basic cases discussed by **parteeRooth**, with a single (polymorphic) entry for *and*, as illustrated below:[24]

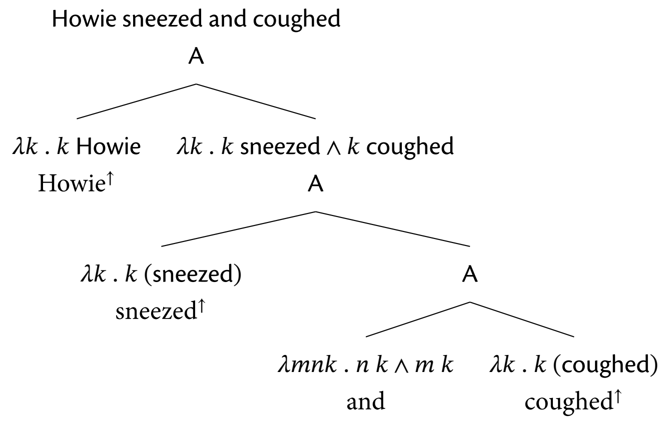
[24] Unlike other lexical entries we've seen so far, *and* is *lexically specified* as seeking continuized arguments. As such, if a value isn't already typed as an instantiation of $\mathsf{K_t}$ a, it must be lifted.

In the derivations below, you can observe that, unlike the cases we've encountered so far, *and* composes with its arguments via **fa!** (**FA!**) rather than **SFA!**.

(57) Lan and some woman arrived.


Lan arrived ∧ ∃x[woman x ∧ arrived x]
A
λk . k Lan ∧ ∃x[woman x ∧ k x]
A
arrived
λk . k Lan
Lan↑
λnk . n k ∧ ∃x[woman x ∧ k x]
A
λmnk . n k ∧ m k
and
λk .∃x[woman x ∧ k x]
some woman

(58) Howie sneezed and coughed.



5.1 *The scope of con/dis-junction*

In a lot of the cases we've seen so far, teasing apart the power of QR from the power of continuations hasn't been at all trivial. In this section, we'll see an example of a case where it's clearer that continuations can help us in ways that QR can't.

Both conjunction and disjunction exhibit "scope" ambiguities. This is illustrated below for conjunction:

(59) You're not allowed to dance and sing.

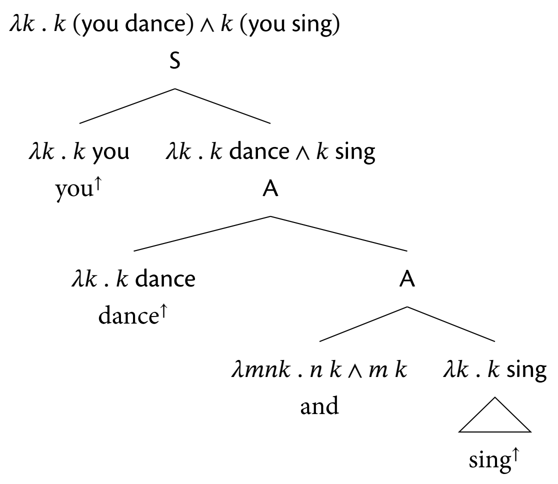
a. *You're not allowed to dance and you're not allowed to sing*
$\wedge > \neg > \Diamond$

b. *You're not allowed to both dance and sing (at the same time)*
$\neg > \Diamond > \wedge$

We can account for the wide/narrow scope ambiguity as a matter of where we LOWER.

Let's first compute the semantic value of the prejacent of the modal:

(60)

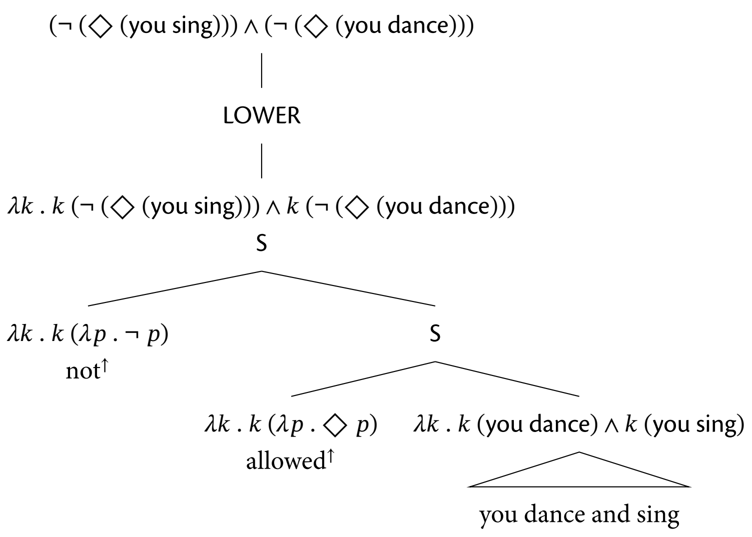

$\lambda k\,.\,k$ (you dance) $\wedge$ k (you sing)
S
$\lambda k\,.\,k$ you
you$^{\uparrow}$
$\lambda k\,.\,k$ dance $\wedge$ k sing
A
$\lambda k\,.\,k$ dance
dance$^{\uparrow}$
A
$\lambda mnk\,.\,n\,k \wedge m\,k$
and
$\lambda k\,.\,k$ sing
sing$^{\uparrow}$

If we LOWER immediately we're just going to get a proposition, which allowed will take as its argument, deriving the narrow scope reading.[25]

[25] Instead of composing via S and lowering, we could equivalently compose lifted *you* with its complement via A, since it's a subject.

The "wide scope" reading is more interesting. We can simply defer lowering, and compose the prejacent with lifted *allowed* via S.

(61)


$(\neg\,(\Diamond\,(\text{you sing})))\wedge(\neg\,(\Diamond\,(\text{you dance})))$
LOWER
$\lambda k\,.\,k\,(\neg\,(\Diamond\,(\text{you sing})))\wedge k\,(\neg\,(\Diamond\,(\text{you dance})))$
S
$\lambda k\,.\,k\,(\lambda p\,.\,\neg\,p)$
$\text{not}^{\uparrow}$
S
$\lambda k\,.\,k\,(\lambda p\,.\,\Diamond\,p)$
$\text{allowed}^{\uparrow}$
$\lambda k\,.\,k\,(\text{you dance})\wedge k\,(\text{you sing})$
you dance and sing

We make the nice prediction that "wide scope" readings of conjunction should be subject to scope islands:

(62) John isn't allowed to claim [that you sing and dance]. ✗ $\wedge > \neg > \Diamond$

5.2 *Split scope with conjunction*

hirsch2017 claims that conjunction reduction is necessary in order to derive the "split scope" reading of the following sentence:

(63) John refused to visit any city in Europe and any city in Asia.
John refused to visit any city in Europe, and he refused to visit any city in Asia. $\wedge >$ refuse $>$ any

hirsch2017's analysis:

(64) John [$_{\text{vP}}$ refused to visit any city in Europe] and
[$_{\text{vP}}$ refused to visit any city in Asia]

Here, I'm going to argue that we can get the split scope reading of (**??**) using just the machinery we've already introduced. Concretely:

- Our entry for *and*.
- The free availability of *external lift*.

First of all, each conjunct is going to be *externally lifted*.

(65) $[\![\text{any city in Europe}]\!]^{\uparrow} = \lambda l \,.\, l\,(\lambda k \,.\, \exists x[\text{city-europe } x \wedge k\ x])$

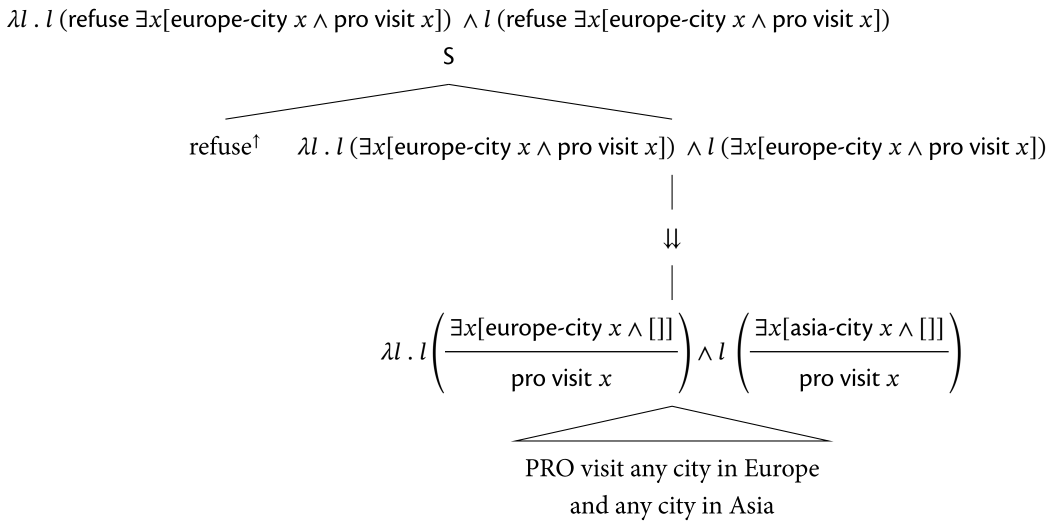
and, in our view, is polymorphic – it's looking for two arguments of type $\mathsf{K}_t\ \mathsf{a}$. This means it can compose the externally lifted DPs, returning the following meaning:

(66) $$\lambda l \,.\, l \left(\frac{\exists x[\text{europe-city } x \wedge [\,]]}{x}\right) \wedge l \left(\frac{\exists x[\text{asia-city } x \wedge [\,]]}{x}\right) : \mathsf{K}_t\ (\mathsf{K}_t\ \mathsf{a})$$

Remember that S_2 is just like S, only it does S of the wrapped values. Composition can proceed via $\uparrow_2$ and S_2 up to the prejacent of *refuse*.

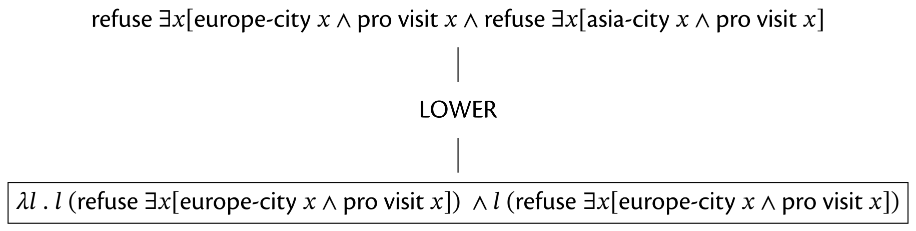
To get the scope of the quantifiers right, we do *internal lower* just before we compose with *refuse*.

(67)


$\lambda l \,.\, l\,(\text{refuse } \exists x[\text{europe-city } x \wedge \text{pro visit } x]) \wedge l\,(\text{refuse } \exists x[\text{europe-city } x \wedge \text{pro visit } x])$
S
$\text{refuse}^{\uparrow}$
$\lambda l \,.\, l\,(\exists x[\text{europe-city } x \wedge \text{pro visit } x]) \wedge l\,(\exists x[\text{europe-city } x \wedge \text{pro visit } x])$
$\Downarrow$
$\lambda l \,.\, l \left(\frac{\exists x[\text{europe-city } x \wedge [\,]]}{\text{pro visit } x}\right) \wedge l \left(\frac{\exists x[\text{asia-city } x \wedge [\,]]}{\text{pro visit } x}\right)$
PRO visit any city in Europe
and any city in Asia

Lowering the result gives us...the split-scope reading:

(68)


$\text{refuse } \exists x[\text{europe-city } x \wedge \text{pro visit } x \wedge \text{refuse } \exists x[\text{asia-city } x \wedge \text{pro visit } x]$
LOWER
$\boxed{\lambda l \,.\, l\,(\text{refuse } \exists x[\text{europe-city } x \wedge \text{pro visit } x]) \wedge l\,(\text{refuse } \exists x[\text{europe-city } x \wedge \text{pro visit } x])}$

So, continuations can get split-scope readings of conjunction straightforwardly. This is a notable result.[26]

[26] To my knowledge, this is a novel observation.

hirsch2017 doesn't consider an analysis of this data in terms of continuations, but does entertain a similar analysis involving lifting the quantificational conjuncts, alongside QR with higher-type traces. **hirsch2017** rejects this analysis on the basis that it allows for the quantifiers to be syntactically above the intensional verb, while semantically reconstructing below it – a configuration which has been argued to be ruled out. Note that an analysis in terms of continuations doesn't face this objection – at no point in the analysis did we need to invoke covert movement.

Note that this theory of split-scope coordination explains the lack of split-scope reading in the following example, from **parteeRooth**.

(69) John hopes [that some company will hire a maid and a cook].
✗ *John hopes that some company will hire a maid,*
and John hopes that some company will hire a cook.

This is simply because the embedded finite clause is a scope island, and therefore the scope of *and* is trapped.

6 *Extension i: DP internal composition and indexed continuations*

As you'll probably have noticed, we've spent this whole time treating quantificational DPs such as *every boy* as primitives.

At this point a natural question to ask is: how do determiners compose with their restrictors?

Surprisingly, the answer isn't as straightforward as you might think.

Naively, we may assume that determiners receive they're standard meaning – essentially, a function from a predicate to a *continuized* individual.

(70) $$[\![\text{every}]\!] \overset{?}{:=} \lambda P \,.\, \frac{\forall y[P\ y \rightarrow [\,]]}{y}$$

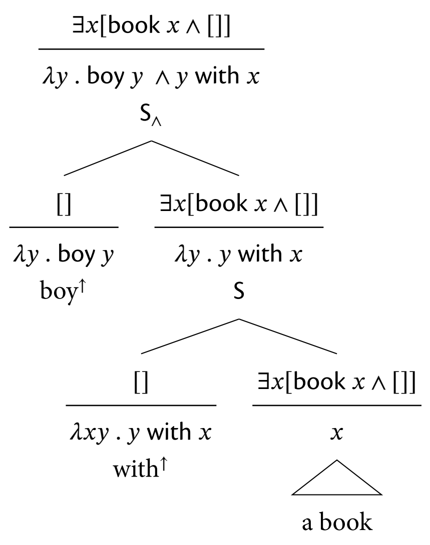
This will (obviously) work fine for a nominal restrictor.

But, what happens if the restrictor itself contains a quantificational expression? Consider the following example:

(71) Every boy with a book left. $\forall > \exists$

Let's first compute the meaning of the restrictor *boy with a book*:[27]

[27] Here, I'm using $S_\wedge$ as an abbreviation for *scopal predicate modification*. In fact, we can lift any binary operation into its scopal counterpart.

(72)


What we end of with is a continuized predicate of type $\frac{\mathtt{t}}{\mathtt{e} \rightarrow \mathtt{t}}$.

How do we compose this with our determiner of type $(\mathtt{e} \rightarrow \mathtt{t}) \rightarrow \frac{\mathtt{t}}{\mathtt{e}}$?

One possiblity is to simply lift the determiner. There are two problems with this approach.

- This will give *a book* scope over *every* – here, we're interested in the surface scope reading.
- Despite potentially being a useful strategy for deriving inversely-linked interpretations, this will ultimately allow the inner quantifier to take scope outside of the containing DP – this runs into issues with Larson's generalization. We'll come back to this.[28]

[28] Essentially, there is some quite strong evidence that DP is a scope island.

Instead, we're going to pursue the idea that the determiner itself *takes scope*.[29]

[29] This has many precedents in the literature. See, for example, **heim1982**.

Consider our type constructor $\mathtt{K}_\mathtt{t}$ – it takes a type $\mathtt{a}$ and returns a new type $(\mathtt{a} \rightarrow \mathtt{t}) \rightarrow \mathtt{t}$.

In principle, we could parameterize $\mathtt{K}$ to any type $\mathtt{r}$ (here $\mathtt{r} = \mathtt{t}$). Let's call $\mathtt{r}$ the *return type*, since it tells us the type of the value we get when the continuation argument k applies to its argument.

What if applying k to its argument gives back an intermediate result of type $\mathtt{i}$, which is subsequently transformed into a final result of type $\mathtt{r}$? We can model this idea using the type constructor $\mathsf{K}_{\mathtt{r}}^{\mathtt{i}}$:[30]

(73) $\mathsf{K}_{\mathtt{r}}^{\mathtt{i}}\ \mathtt{a} := (\mathtt{a} \to \mathtt{i}) \to \mathtt{r}$

[30] This ultimately goes back to **wadler1994**.

barkerShan2015 generalize tower notation to the more general type-schema.[31]

[31] See also **Charlowc**.

(74) Tripartite tower types (def.)

$$\frac{\mathtt{r} \mid \mathtt{i}}{\mathtt{a}} := (\mathtt{a} \to \mathtt{i}) \to \mathtt{r}$$

We can think of our existing tower notation as an abbreviation for a tripartite tower type, where the intermediate and final result types happen to be the same:

(75) Bipartite towers as abbreviations for tripartite towers

$$\frac{\mathrm{r}}{\mathrm{a}} := \frac{\mathrm{r} \mid \mathrm{r}}{\mathrm{a}}$$

Now that we have tripartite tower types, we can think of the restrictor argument c of *every* as a *continuation argument*.

(76) Standard determiner semantics for *every*

$$[\![\text{every}]\!] := \lambda c\,.\left[\lambda P\,.\frac{\forall y[P\,y \to [\,]]}{y}\right](\lambda x\,.\,c\,x) \qquad (\mathtt{e} \to \mathtt{t}) \to \frac{\mathtt{t}}{\mathtt{e}}$$

We can abbreviate the meaning of *every* as a tower, where c is the continuation argument:

(77)
$$\frac{\left[\lambda P\,.\frac{\forall y[P\,y \to [\,]]}{y}\right](\lambda x\,.\,[\,])}{x} \quad : \frac{\frac{\mathtt{t}}{\mathtt{e}} \mid \mathtt{t}}{\mathtt{e}}$$

Our existing definition of S can be made more type-general, in order to accommodate tripartite tower types. *Adjacent types* match and cancel out:

(78)
$$\mathsf{S} : \frac{\mathrm{r} \mid \mathrm{i}}{\mathrm{a} \to \mathrm{b}} \to \frac{\mathrm{i} \mid \mathrm{j}}{\mathrm{a}} \to \frac{\mathrm{r} \mid \mathrm{j}}{\mathrm{b}}$$

The actual definition of S doesn't change.

(79) *scopal function application* (def.)
$m \mathsf{S} n \coloneqq \lambda k \,.\, m\,(\lambda x \,.\, n\,(\lambda y \,.\, k\,(x \mathsf{A} y)))$

Likewise, the type of *lower* is further generalized; the definition doesn't change:

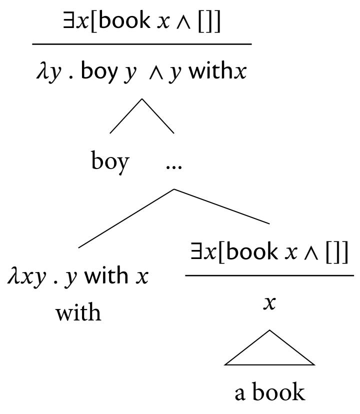
(80) $(\downarrow) : \dfrac{\mathsf{a} \mid \mathsf{b}}{\mathsf{b}} \to \mathsf{a}$

We can now make sense of the idea that the determiner *takes scope*. Let's consider again the following complex DP:

(81) Every boy with a book.

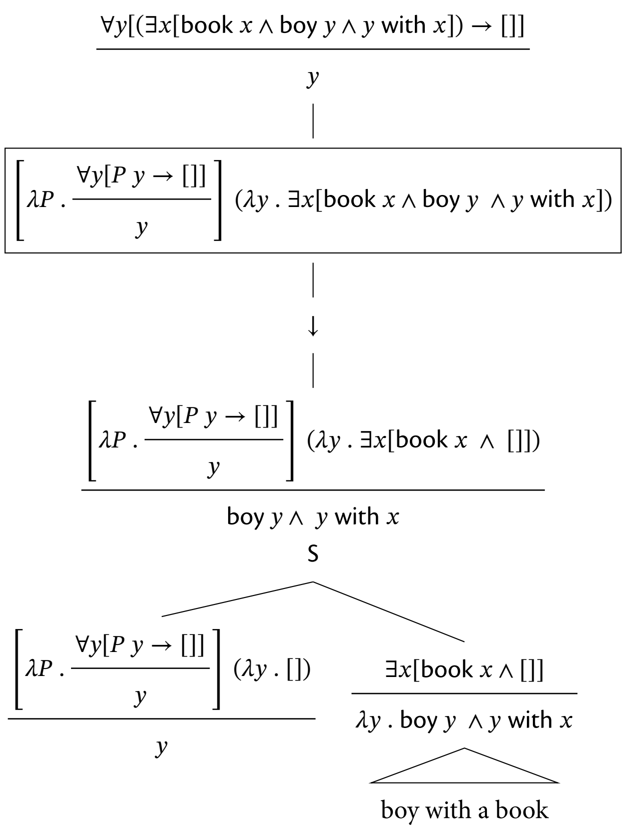
First, we compute the meaning of the restrictor, returning something of type $\dfrac{\mathsf{t} \mid \mathsf{t}}{\mathsf{e} \to \mathsf{t}}$:

(82) Step 1: compute *boy with a book*



Recall that *every* is of type $\dfrac{\frac{\mathsf{t}}{\mathsf{e}} \mid \mathsf{t}}{\mathsf{e}}$, this is something that can compose with the restrictor via our more type-general formulation of **sfa!**.

(83)


∀y[(∃x[book x ∧ boy y ∧ y with x]) → []]
y
[λP . ∀y[P y → []] / y] (λy . ∃x[book x ∧ boy y ∧ y with x])
↓
[λP . ∀y[P y → []] / y] (λy . ∃x[book x ∧ []])
boy y ∧ y with x
S
[λP . ∀y[P y → []] / y] (λy . [])
y
∃x[book x ∧ []]
λy . boy y ∧ y with x
boy with a book

Exercise

As homework, you can attempt to work out the above derivation yourself. Convince yourself that the types work out.

7 *Extension ii: Exceptional scope of indefinites*

Unlike other quantificational expressions, indefinites don't obey scope islands:

(84) Roger hopes that some student will be absent. $\exists >$ hope

Charlowc outlines a theory of exceptional scope framed in terms of continuation semantics, compatible with the assumption that scope islands are obligatorily evaluated.

Charlowc assumes that, unlike other quantificational expressions, indefinites denote sets of individuals:

(85) a. $[\![\text{some student}]\!] := \{\, x \mid \text{student } x \,\}$
b. $[\![\text{some student}]\!] : \{\, \texttt{e} \,\}$

The central component of **Charlowc**'s proposal is a method (let's call it ⋆) for lifting a set into a scope taker.[32]

(86) $m^{\star} := \lambda k \,.\, \bigcup_{x \in m} k\, x \qquad \star : \{\,\texttt{e}\,\} \rightarrow (\texttt{e} \rightarrow \{\,\texttt{t}\,\}) \rightarrow \{\,\texttt{t}\,\}$

[32] If you're familiar with the notion of *monads*, you'll noticed that ⋆ is just the *bind* operation of the set monad.

One of **Charlowc**'s insights is that, for any monad M, the bind of M provides a way of lifting an inhabitant of $M\ a$ into a scope taker.

In fact, we can write ⋆ using tower notation:

(87) $m^{\star} := \dfrac{\bigcup_{x \in m} \; []}{x}$

As you'll notice, ⋆ gives back a *continuized individual*, but with the return type generalized to $\{\,\texttt{t}\,\}$ rather than $\texttt{t}$.

Our existing definitions of lift and **sfa!** allow us to compose a ⋆-shifted indefinite straightforwardly, once the return type is set to $\{\,\texttt{t}\,\}$. The actual definitions of the operations don't change.

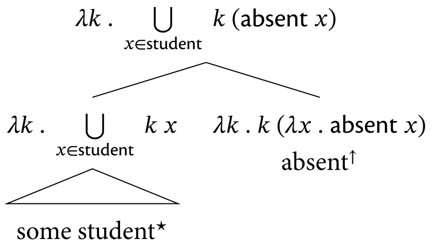
Here I'll show how ⋆, once combined with our existing mechanisms for doing scope-taking – namely lift and S – *predicts* that indefinites can exceptionally scope out of scope islands.

I'll demonstrate this for our example: *Roger hopes that some student is absent*:

(88) Step 1: lift *some student* into a scope-taker

$$\begin{tikzcd}
\lambda k \,.\, \bigcup_{x \in \text{student}} k\, x \arrow[d, dash] \\
\star \arrow[d, dash] \\
\{\, x \mid \text{student } x \,\} \\[-2em]
\text{some student}
\end{tikzcd}$$

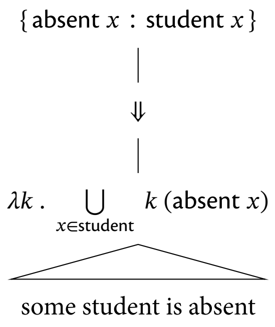
(89) Step 2: scope via lift and S


$\lambda k . \bigcup_{x \in \text{student}} k\,(\text{absent}\ x)$
$\lambda k . \bigcup_{x \in \text{student}} k\,x$
$\lambda k . k\,(\lambda x . \text{absent}\ x)$
absent$^{\uparrow}$
some student*

We've reached a scope island now, so we need to *evaluate* it. We can't use our ordinary lower function – since we're dealing with sets, we lower by feeding n Partee's IDENT, rather than the identity function:

(90) $m^{\Downarrow} \coloneqq m\,(\lambda x . \{\, x \,\})$ $\qquad\qquad \Downarrow : ((\mathsf{a} \to \{\, \mathsf{a} \,\}) \to \{\, \mathsf{a} \,\}) \to \{\, \mathsf{a} \,\}$

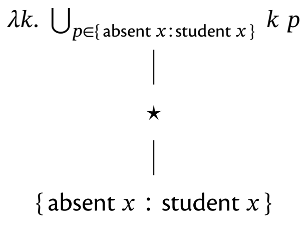
(91) Step 3: evaluate the scope island:


$\{\, \text{absent}\ x : \text{student}\ x \,\}$
$\Downarrow$
$\lambda k . \bigcup_{x \in \text{student}} k\,(\text{absent}\ x)$
some student is absent

What we end up with is a set of propositions of the form *x is absent*, where *x* is a student.

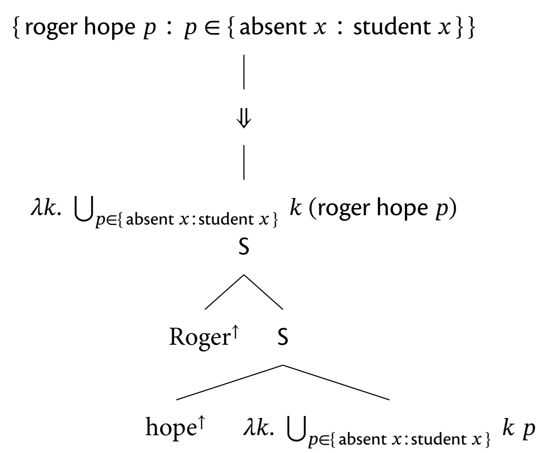
Now we re-lift the scope island into a scope taker via ⋆:

(92) Step 4: re-lift via ⋆


$\lambda k. \bigcup_{p \in \{\, \text{absent}\ x : \text{student}\ x \,\}} k\ p$
$\star$
$\{\, \text{absent}\ x : \text{student}\ x \,\}$

Now we scope out the scope-island via lift and S – the indeterminacy introduced by the indefinite survives, and we get exceptional scope.

(93) Step 5: scope via lift and $\star$:


$\{\,\text{roger hope}\ p : p \in \{\,\text{absent}\ x : \text{student}\ x\,\}\}$
$\Downarrow$
$\lambda k.\ \bigcup_{p \in \{\,\text{absent}\ x : \text{student}\ x\,\}} k\ (\text{roger hope}\ p)$
S
Roger$^{\uparrow}$
S
hope$^{\uparrow}$
$\lambda k.\ \bigcup_{p \in \{\,\text{absent}\ x : \text{student}\ x\,\}} k\ p$

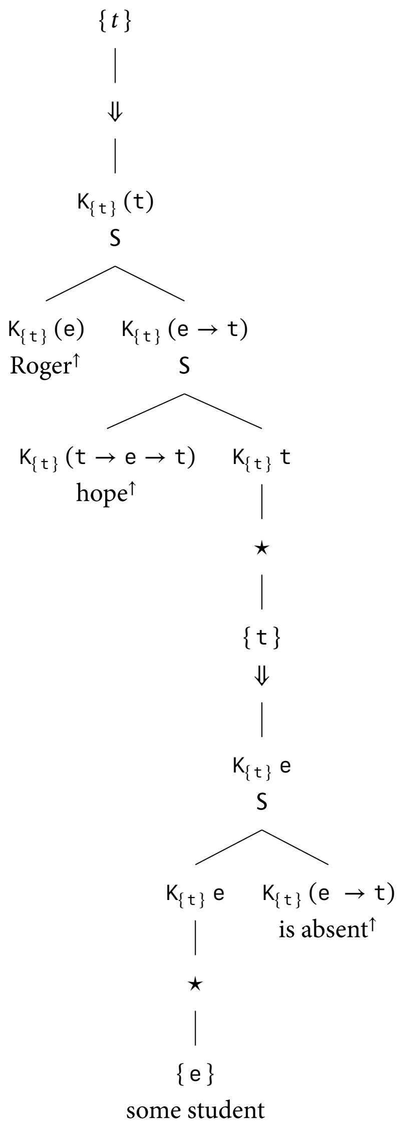
The result of lowering is equivalent to the following:

(94) $\{\,\text{roger hope (absent x) : student x}\,\}$

In brief, unlike other quantifiers, indefinites introduce indeterminacy (which we model as a set). Indeterminacy survives obligatory evaluation at a scope island.

Here is the derivation again from a bird's eye view:

(95)


{t}
⇓
K{t} (t)
S
K{t} (e)
Roger↑
K{t} (e → t)
S
K{t} (t → e → t)
hope↑
K{t} t
⋆
{t}
⇓
K{t} e
S
K{t} e
K{t} (e → t)
is absent↑
⋆
{e}
some student