PATRICK D. ELLIOTT

SCOPE PROJECTION & EVALUATION ORDER

Contents

Introduction

Continuation semantics

0.1 Why continuations?

In the first part of these notes, we'll be introducing a unified approach to scope-taking in natural language developed primarily by Chris Barker and Chung-chieh Shan: *continuation semantics*.[1] I'd like to start out by saying something about the following question: *why bother?*. If you're a certain kind of person (and, I think it's fair to say, *I'm* this kind of person), the following answer should be sufficient: *playing with new toys is fun*. There is however, also a longer answer.

[1] See, e.g., Barker 2002, Shan 2002, Shan 2005, and Barker & Shan 2014.

In these notes, to a large extent we're going to be investigating mechanisms via which meanings *get high*. To put this in a different way, expressions of natural language come with conventionalized meaning components. Often, meaning components end up interpreted in places that don't necessarily correspond to the pronounced position of the expression they're associated with. Special cases of this include, but are almost certainly not limited to:

- Scope-taking (the focus of the first part of this seminar).

- Presupposition projection (the focus of the latter part of this seminar).

- Other varieties of "projective" content (e.g., expressives, appositives, etc.).

We have fairly well-established techniques for dealing with these phenomena, e.g., Quantifier Raising (QR) for scope[2], trivalence for presupposition projection, and conventional implicature for expressives and appositives.[3] *Continuations* provide a powerful abstraction for modeling meaning components "getting high" in a more general, uniform way.[4] This doesn't automatically make them preferable, but if we can get away with reducing our set of theoretical primitives while maintaining the same empirical coverage, we should at least pursue the possibility.

[2] May 1977, Heim & Kratzer 1998, etc.

Add citations

[3] Potts 2005, McCready 2010, Gutzmann 2015, etc.

[4] This goes beyond just *meaning*, and can be generalized to, e.g., phonological and syntactic features. See Elliott 2019b.

Continuations were originally developed[5] by computer scientists in order to account for computations that are, in some sense, *delayed* until later on. As such, they're especially well-suited to modelling meaning components, the evaluation of which is *delayed* until later in the derivation. Continuations have been applied to a broad range of natural language phenomena. Some of the work we'll discuss in these notes includes:

[5] Or rather, *discovered*. See, e.g., Danvy & Filinski (1992) and Wadler (1994) for important foundational work on *delimited continuations* – the variety we'll be discussing in this course.

- Barker & Shan's (and related works) unified theory of scope-taking *scope taking*.

- Grove's (2019) analysis of presupposition projection as a *scopal* phenomenon, using continuations.

- de Groote's (2006) application of continuations to develop a *compositional* dynamic semantics (another important reference is Charlow 2014).

- My recent work using continuations to model non-local readings of expressive adjectives (Elliott 2019a) and overt movement (Elliott 2019b)

One of the *design features* of continuation semantics that is of special interest to us is its *built-in left-to-right* bias. One thing we'll be thinking about in some depth is how meaning that "gets high", such as scope-takers, interact with anaphoric expressions and other dependees. As has been observed for quite some time, interaction between scope and anaphora exhibits a *leftness bias*.

(1) Every student^i's mother adores their$_i$ advisor. (2) *Their$_i$ advisor adores every student^i's mother.

The contrast above is an example of a Weak Crossover (wco) violation. Roughly, weak crossover can be stated as follows:

(3) Weak Crossover (wco)
     Scope may feed binding *unless* the bound expression *precedes* the scope-taker.

Later in these notes, I'll explore the possibility, tentatively, that weak crossover isn't just about quantificational scope, but parallel effects can be observed with projective meanings more generally. I'll give you a couple of examples here to whet your appetite, although we won't get back to this until quite a bit later in the course. First, I'd like to suggest that *presupposition projection can feed anaphora*:

(4)   Daniel doesn't know Paul has a sister^x, although he has seen her$_x$.

Focus on the construal where the indefinite takes scope below the intensional verb. Nevertheless, anaphora is possible. The natural way to think about this is as presupposition projection feeding anaphora:[6]

(5) | Paul has a sister | Daniel doesn't know Paul has a sister, although he has seen her.

Having established that presupposition projection can feed anaphora, let's see what happens when the pronoun is made to *precede* the presupposition trigger:

(6)   #Her$_x$ boss doesn't know that Paul has a sister^x.

Again, focus on the construal where the indefinite takes scope below the intensional verb. Our example doesn't have the reading indicated. This is surprising given that (a) presuppositions project, (b) presupposition projection can feed anaphora. Our hunch is that, what is to blame here is a more general form of WCO – *projective meaning may feed anaphora unless the anaphor precedes the projector.* Continuations are sufficiently general, that we can use them to model a wide variety of phenomena. Since they come with a built in linear bias[7], they seem like a prime candidate for modeling linear biases more generally.

> Keep converting to prose from here

0.2   Plan

This week:

- We're going to develop an understanding of Barker & Shan's tower notation, and how to translate from towers to flat representations, and vice versa.

- We'll get a handle on continuation semantics' linear bias, encoded in the composition rules themselves.[8]

- We'll show how to account for scope ambiguities via a combination of *lowering* and multi-story towers.

- Scope islands in terms of obligatory evaluation.

[6] A cautionary note: surprisingly, this doesn't follow from standard dynamic theories (such as Heim 1982, Beaver 2001), since presuppositions are themselves static. A natural move is to instead treat presuppositions as themselves dynamic statements; see Elliott & Sudo (2019) for a theory with this character.

[7] And, indeed, arguably one of the most empirically successful accounts of WCO is couched in terms of continuation semantics – see Shan & Barker (2006). We'll discuss this in several weeks time.

[8] This will be necessary preparation for our discussion of Shan & Barker's (2006) theory of crossover.

Next week:

- Using continuations to account for generalized con/dis-junction.

- Indexed continuations and a compositional semantics for determiners.

- Continuations and exceptionally scoping indefinites (Charlow 2014).

If we have time:

- Antecedent Contained Deletion.

- Extraposition and scope.

- Expressive adjectives.

0.3 Some notation conventions

Generally speaking, I'll be assuming Heim & Kratzer 1998 as background, but I'll depart from their notation slightly.

Expressions in the meta-language will be typeset in sans serif.

$$[\![[_{\text{DP}} \text{ John}]]\!] := \underbrace{\text{John}}_{\text{individual}}$$

Seeing as its primitive, we'll treat white-space as function application, e.g.:

(7) $(\lambda x \,.\, \text{left } x) \text{ paul} = \text{left paul}$

Function application associates to the *left*:

(8) $(\lambda x \,.\, \lambda y \,.\, y \text{ likes } x) \text{ paul sophie} \equiv ((\lambda x \,.\, \lambda y \,.\, y \text{ likes } x) \text{ paul}) \text{ sophie}$

Question

- -

Can the following expression be reduced?

(9) $(\lambda k \,.\, \exists z[k\ z]) \,(\lambda x \,.\, \lambda y \,.\, y \ \text{likes } x) \text{ Sam}$

We'll write *types* in a `fixed width font`. We have our familiar primitive types...

(10) type := e | t | s | ...

...and of course *function types.* Unlike Heim & Kratzer (1998), who use ⟨.⟩ as the constructor for a function type, we'll be using the (more standard (outside of linguistics!)) arrow constructor (→):

(11) ⟨e, t⟩ ≡ e → t

The constructor for function types *associates to the right*:

(12) e → e → t ≡ e → (e → t)

> Question
> -
> Which of the following is the correct type for a quantificational determiner?
>
> (13) ⟦every⟧ : (e → t) → (e → t) → t
>
> (14) ⟦every⟧ : e → t → (e → t) → t

0.4 The Partee triangle

The purpose of this section is to show that continuation semantics is, in a certain sense, *already implicit* in the inventory of type-shifters standardly assumed in formal semantics.

(15) LIFT $x := \lambda k \, . \, k \, x$ $\qquad\qquad$ LIFT : e → (e → t) → t

(16) IDENT $x := \lambda y \, . \, y = x$ $\qquad\qquad$ IDENT : e → e → t

(17) BE $Q := \lambda x \, . \, Q \, (\lambda y \, . \, y = x)$ $\qquad$ BE : ((e → t) → t) → e → t

Evidence for IDENT:

(18) This man is John.

Evidence for BE:

(19) This man is an authority on heavy metal.

We'll come back to LIFT in a moment.

> **Commutative diagrams**
> -
> The *Partee triangle* is a *commutative diagram*. We say that a diagram
> *commutes* if, when there are multiple paths between two points, those
> paths are equivalent. The equivalence in (21) is therefore expressed by
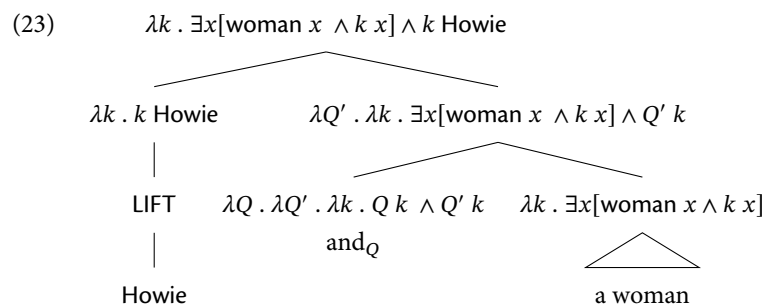> the triangle.

(20) The Partee triangle[9]

$$e \xrightarrow{\text{LIFT}} (e \rightarrow t) \rightarrow t$$
$$\text{IDENT} \searrow \swarrow \text{BE}$$
$$e \rightarrow t$$
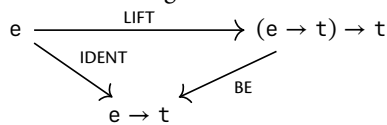
(21) ident $\equiv$ BE $\circ$ ($\uparrow$)

OBSERVATION: quantificational and non-quantificational DPs can be coordinated:

(22) [Howie and a woman] entered the club

LIFT allows something that, by virtue of its quantificational nature, is an *inherent* scope-taker, to combine with something that *isn't*:[10]

[10] If you're familiar with Partee & Rooth 1983 you'll notice that the *and* that coordinates quantificational DPs (written here as and_Q), is just the result of applying their *generalized conjunction* rule. We'll return to generalized conjunction, and the connection to continuations next week.

(23) $\lambda k \, . \, \exists x[\text{woman } x \, \wedge k \, x] \wedge k \text{ Howie}$

$\lambda k \, . \, k \text{ Howie}$ $\lambda Q' \, . \, \lambda k \, . \, \exists x[\text{woman } x \, \wedge k \, x] \wedge Q' \, k$

LIFT $\lambda Q \, . \, \lambda Q' \, . \, \lambda k \, . \, Q \, k \, \wedge Q' \, k$ $\lambda k \, . \, \exists x[\text{woman } x \wedge k \, x]$

and_Q

Howie   a woman

0.4.1   Generalizing the triangle

Based on the way in which LIFT and friends are defined, in theory we could replace e with *any* type. Let's give a more general statement of LIFT and friends as polymorphic functions:
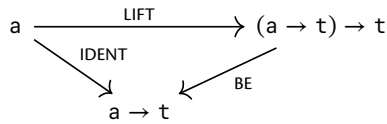
(24)   LIFT $x := \lambda k \ . \ k \ x$ $\qquad\qquad\qquad$ LIFT : $a \to (a \to t) \to t$

(25)   IDENT $x := \lambda y \ . \ y = x$ $\qquad\qquad\qquad$ IDENT : $a \to a \to t$

(26)   BE $Q := \lambda x \ . \ Q \ (\lambda y \ . \ y = x)$ $\qquad\qquad$ BE : $((a \to t) \to t) \to a \to t$

The diagram, of course, still commutes:

(27)   The (generalized) Partee triangle

$$a \xrightarrow{\text{LIFT}} (a \to t) \to t$$

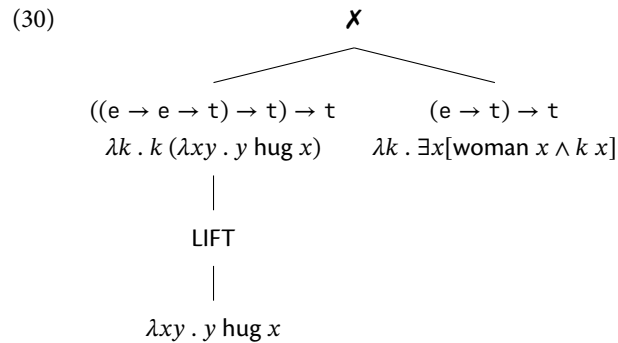IDENT $\qquad$ BE

$$a \to t$$

Why might a polymorphic LIFT be useful? Recall that we used a typed instantiation of LIFT in order to allow a quantificational thing to combine with a non-quantificational thing. Polymorphic LIFT allows us to type-lift, e.g., a function that takes multiple arguments:

(28)   LIFT $(\lambda xy \ . \ y \text{ hug } x) = \overbrace{\lambda k \ . \ k \ (\lambda xy \ . \ y \text{ hug } x)}^{((e \to e \to t) \to t) \to t}$

One tantalizing possibility is that this allow us to combine a non-quantificational transitive verb with a quantificational DP.

(29)   Howie ⏐hugged a woman⏐

(30)

✗

$((e \to e \to t) \to t) \to t$ $\qquad$ $(e \to t) \to t$
$\lambda k \ . \ k \ (\lambda xy \ . \ y \text{ hug } x)$ $\quad$ $\lambda k \ . \ \exists x[\text{woman } x \wedge k \ x]$

LIFT

$\lambda xy \ . \ y \text{ hug } x$

Unfortunately, assuming the usual inventory of composition rules (i.e., *function application*, *predicate modification*, and *predicate abstraction*), we're stuck.[11] So, **let's invent a new one**.

Here's the intuition we're going to pursue. Let's look again at the types. One way of thinking about what LIFT as follows: it takes an a-type thing and adds a "wrapper". Quantificational DPs, on the other hand come "pre-wrapped".

- LIFT $[\![\text{hug}]\!]$: $(\boxed{(e \to e \to t)} \to t) \to t$

- $[\![\text{a woman}]\!]$: $(\boxed{e} \to t) \to t$

If we look at the wrapped-up types, we see a *function* from individuals, and an individual – namely, two things that can combine via function application.

What we need to accomplish is the following:

- Unwrap lifted *hug*.

- Unwrap *a woman*.

- Use function application to combine the unwrapped values.

- Finally, wrap the result back up! Think of the quantificational meaning as being like a taco – it isn't really a taco without the wrapper, therefore we don't want to throw the wrapper away.

In order to accomplish this, we'll define a new composition rule: Scopal Function Application (SFA). We're going to define SFA in terms of Function Application (FA); we haven't been explicit about how FA is defined yet, so let's do that now. We'll write FA as the infix operator A.[12]

(31)   Function Application (FA) (def.)
   a. $f \text{ A } x := f\ x$ $A : (a \to b) \to a \to b$
   b. $x \text{ A } f := f\ x$ $A : a \to (a \to b) \to b$

> **Scopal Function Application (SFA)**
>
> -
>
> We'll write SFA as the infix operator S. Note that, since A is overloaded already, and S is defined in terms of A, S gets overloaded too.
>
> (32) Scopal Function Application (SFA) (def.)
>
> m S $n := \lambda k \,.\, m \,(\lambda a \,.\, n \,(\lambda b \,.\, k \,(a \text{ A } b)))$
>
> $\qquad$ S : $(((a \to b) \to t) \to t) \to ((a \to t) \to t) \to (b \to t) \to t$
>
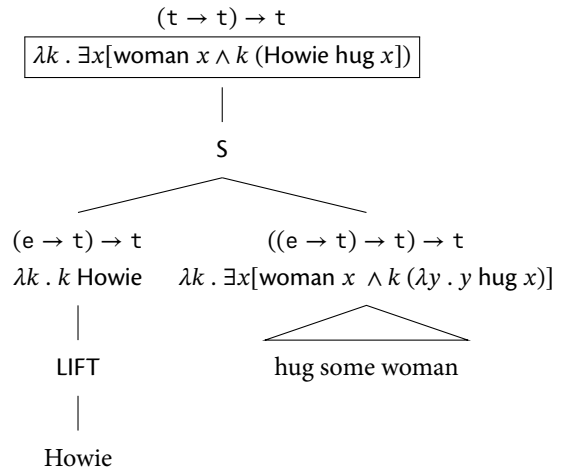> $\qquad$ S : $((a \to t) \to t) \to (((a \to b) \to t) \to t) \to (b \to t) \to t$

Now, let's illustrate how SFA plus generalized LIFT allows to compose a quantificational thing with non-quantificational things, using a simple example sentence:

(33) Howie hugged some woman.

(34) Step 1: compose *some woman* with LIFT-ed *hug*.

$$((e \to t) \to t) \to t$$
$$\boxed{\lambda k \,.\, \exists x[\text{woman } x \ \wedge k \,(\lambda y \,.\, y \text{ hug } x)]}$$

S

$((e \to e \to t) \to t) \to t \qquad\qquad (e \to t) \to t$
$\lambda k \,.\, k \,(\lambda xy \,.\, y \text{ hug } x) \qquad\qquad \lambda k \,.\, \exists x[\text{woman } x \wedge k \, x]$

LIFT $\qquad\qquad\qquad\qquad$ some woman

hug
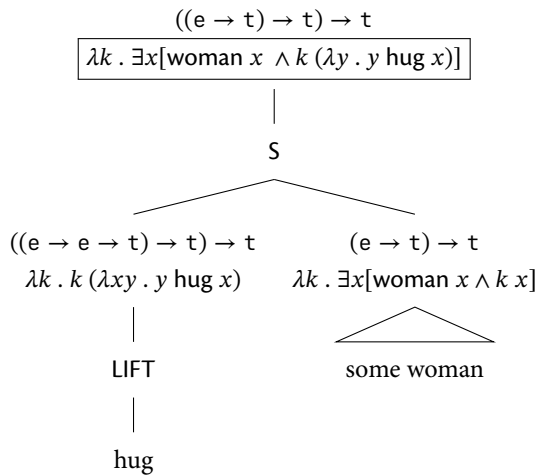
(35) Step 2: compose the resulting VP-denotation with LIFT-ed *Howie*

$$(t \to t) \to t$$
$$\boxed{\lambda k \,.\, \exists x[\text{woman } x \wedge k \,(\text{Howie hug } x])}$$

S

$(e \to t) \to t \qquad\qquad ((e \to t) \to t) \to t$
$\lambda k \,.\, k \text{ Howie} \qquad\qquad \lambda k \,.\, \exists x[\text{woman } x \ \wedge k \,(\lambda y \,.\, y \text{ hug } x)]$

LIFT $\qquad\qquad\qquad\qquad$ hug some woman

Howie

At this point, it's worth mentioning that we've **bootstrapped continuation semantics** from an independently motivated type-lifting operation, plus a natural composition rule SFA; continuations aren't scary at all!

The λk argument is standardly referred to as the *continuation argument* in the derivations above. SFA allows the continuation argument to get "passed up".

We're now tantalizingly close to deriving the right kind of object for the sentential meaning (namely, something of type t). Only, what we have is something

of type ($\boxed{t} \rightarrow t) \rightarrow t$, i.e., a truth value in a "wrapper". How do we get back the wrapped up value? We saturate the k argument with the identity function. We'll call this operation LOWER.[13]

(36) LOWER (def.)

LOWER $m := m\ id$ LOWER : $((a \rightarrow a) \rightarrow a) \rightarrow a$

Applying LOWER to the final value in (35) gives us a type t sentential meaning.
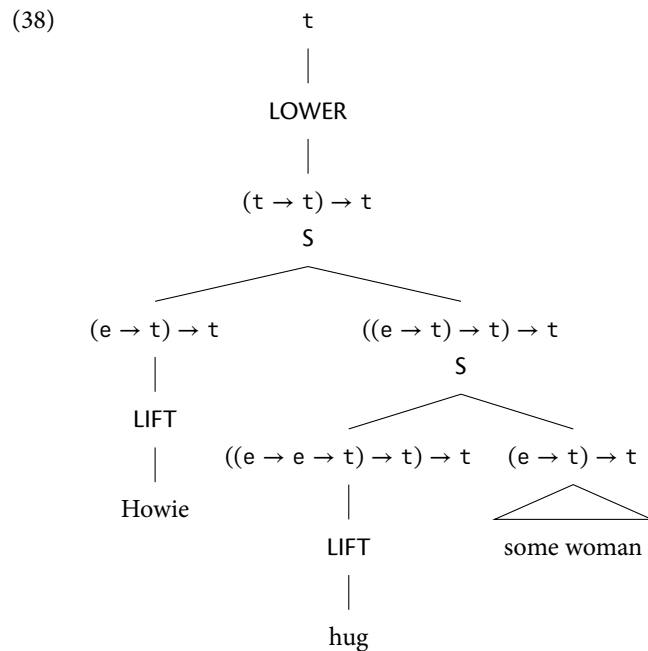
(37) LOWER ($\lambda k\ .\ k\ (\exists x[\text{Howie hug } x])$)

$= (\lambda k\ .\ k(\exists x[\text{Howie hug } x]))\ id$ by def.

$= id\ (\exists x[\text{Howie hug } x])$ reduce

$= \exists x[\text{Howie hug } x]$ reduce

Let's zoom back out and see how all the pieces fit together by looking at the *graph of the derivation*.

(38)

So far, we've provided an account of how quantificational things compose with non-quantificational things, by making use of...

- ...an independently motivated type-shifting rule (LIFT)...

- …a way to apply LIFT-ed values (S)…

- …and a way to get an ordinary value back from a LIFT-ed value (LOWER).

This seems pretty nice, but as I'm sure you've noticed, things are quickly going to get pretty cumbersome with more complicated sentences, especially with multiple quantifiers. Before we go any further, let's introduce some notational conveniences.

0.5   Towers

We've been using the metaphor of a *wrapper* for thinking about what LIFT does to an ordinary semantic value. Let's make this a bit more transparent by introducing a new *type constructor* for LIFT-ed values.[14]

(39) $K_t\ a := (a \rightarrow t) \rightarrow t$

[14] A *type constructor* is just a function from a type to a new type – here, it's a rule for taking any type a and returning the type of the corresponding LIFT-ed value.

- Quantificational DPs are therefore of type $K_t\ e$ (inherently).

- LIFT takes something of type a, and lifts it into something of type $K_t\ a$.

Rather than dealing with *flat* expressions of the simply-typed lambda calculus, which will become increasingly difficult to reason about, we'll follow Barker & Shan 2014 in using *tower notation*.[15,16]

Let's look again at the meaning of a quantificational DP. The k argument which acts as the *wrapper* is called the *continuation argument*.

(40)   Ordinary quantifier meanings:
$[\![\text{some woman}]\!] := \lambda k\ .\ \exists x[\text{woman } x\ \wedge k\ x]$

(41)   Quantifiers using tower notation:
$[\![\text{some woman}]\!] := \dfrac{\exists x[\text{woman } x \wedge [\,]]}{x}$

(42)   Lifted meanings using tower notation:
$\text{LIFT}\,([\![\text{hug}]\!]) = \dfrac{[\,]}{\lambda xy\ .\ y\ \text{hug } x}$

In general:

[15] To my mind, one of Barker & Shan's central achievements is simply the introduction of an accessible notational convention for reasoning about the kinds of lifted meanings we're using here.

[16] It's important to bear in mind that towers are just *syntactic sugar* for flat lambda expressions; we should always be able to translate back from towers to lambda expressions. Towers have no privileged theoretical status, unlike, e.g., Discourse Representation Structures, but are merely abbreviations.

(43) Tower notation (def.)

$$\frac{f\ []}{x} := \lambda k\ .\ f\ (k\ x)$$

We can use tower notation for types too:

(44) Tower types (def.)

$$\frac{b}{a} := (a \to b) \to b \qquad\qquad\qquad \equiv K_b\ a$$

We can now redefine our type constructor K_t, and our type-shifting operations using our new, much more concise, tower notation. These will be our canonical definitions from now on. We'll also start abbreviating a LIFT-ed value a as $a^\uparrow$ and a LOWER-ed value b as $b^\downarrow$.

(45) The continuation type constructor K_t (def.)

$$K_t\ a := \frac{t}{a}$$

(46) LIFT (def.)[17]

$$a^\uparrow := \frac{[]}{a} \qquad\qquad\qquad (\uparrow) : a \to K_t\ a$$

[17] Thinking in terms of towers, LIFT takes a value a and returns a "trivial" tower, i.e., a tower with an empty top-story.

(47) Scopal Function Application (SFA) (def.)[18]

$$\frac{f\ []}{x}\ S\ \frac{g\ []}{y} := \frac{f\ (g\ [])}{x\ A\ y} \qquad\qquad S : K_t\ (a \to b) \to K_t\ a \to K_t\ b$$

[18] SFA takes two scopal values – one with a function on the bottom floor, and the other with an argument on the bottom floor – and combines them by (i) doing function application on the bottom floor, and (ii) *sequencing* the scope-takers.
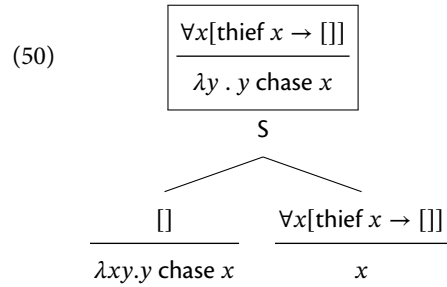
(48) LOWER (def.)[19]

$$\left(\frac{f\ []}{p}\right)^\downarrow = f\ p \qquad\qquad\qquad (\downarrow) : K_t\ t \to t$$

[19] LOWER *collapses the tower*, by saturating the continuation argument with the identity function.
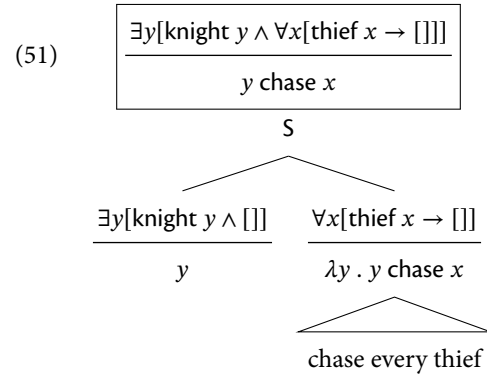
In order to see the tower notation in action, let's go through an example involving multiple quantifiers, and show how continuation semantics derives the surface scope reading:

(49) Some knight chased every thief.   ∃ > ∀
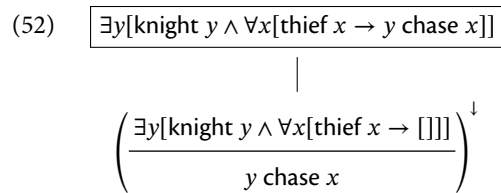
First, we combine *every thief* with lifted *chase* via SFA:
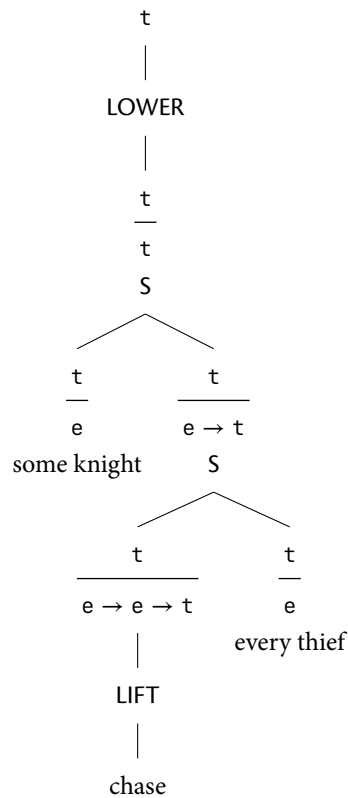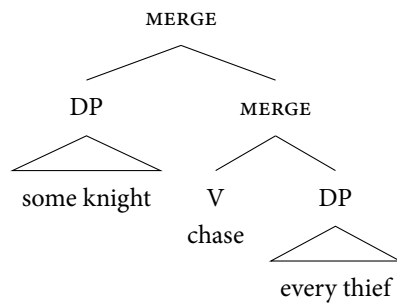
(50)

$$\boxed{\dfrac{\forall x[\text{thief } x \rightarrow []]}{\lambda y \,.\, y \text{ chase } x}}$$

S

$$\dfrac{[]}{\lambda xy.y \text{ chase } x} \qquad \dfrac{\forall x[\text{thief } x \rightarrow []]}{x}$$

Next, the (boxed) VP value combines with *some knight* via SFA:

(51)

$$\boxed{\dfrac{\exists y[\text{knight } y \wedge \forall x[\text{thief } x \rightarrow []]]}{y \text{ chase } x}}$$

S

$$\dfrac{\exists y[\text{knight } y \wedge []]}{y} \qquad \dfrac{\forall x[\text{thief } x \rightarrow []]}{\lambda y \,.\, y \text{ chase } x}$$

chase every thief

Finally, the resulting tower is collapsed via *lower*:

(52) $\boxed{\exists y[\text{knight } y \wedge \forall x[\text{thief } x \rightarrow y \text{ chase } x]]}$

$$\left(\dfrac{\exists y[\text{knight } y \wedge \forall x[\text{thief } x \rightarrow []]]}{y \text{ chase } x} \right)^{\downarrow}$$

Let's zoom out and look at the *graph of the syntactic derivation* alongside the *graph of the semantic derivation*.

MERGE

DP — MERGE

some knight

V DP
chase
every thief

t

LOWER

$\dfrac{t}{t}$

S

$\dfrac{t}{e}$ $\dfrac{t}{e \rightarrow t}$

some knight   S

$\dfrac{t}{e \rightarrow e \rightarrow t}$ $\dfrac{t}{e}$

LIFT   every thief

chase

Type-shifting operations are interleaved with syntactic operations. In order to restrict this system, it seems natural to place an economy constraint on S and LIFT.[20]

(53) Economy condition on type-shifting
      MERGE in the syntax is interpreted as A/S in the semantics, whichever is well-typed. LIFT may apply freely to the extent that it is necessary for the derivation to proceed.

0.6   Scopal ambiguities

0.6.1   Interaction between scope-takers and other operators

Right now, we have a theory which is very good at deriving the surface scope reading of a sentence with multiple quantifiers. It can also derive some ambiguities that arise due to interactions of quantifiers and scopally *immobile* expressions, such as intensional predicates. Consider, e.g., the interaction between a universal quantifier and the desire verb *want*.

(54) Daniele wants to join every group chat.   want $> \forall$; $\forall >$ want

(54) is ambiguous: (i) if *every* takes scope below *want*, it is true if Dani has a desire about joining every group chat, (ii) if *every* takes scope over *want*, it is true if every group chat is s.t. Dani has a desire to join in.

We actually already have everything we need in order to account for this. In a nutshell, the two readings correspond to an application of LOWER either below or above the intensional predicate.[21]

(55) Daniele wants to join every group chat.   want $> \forall$

...

Daniele $\lambda y . y$ wants $(\forall x[\text{groupChat } x \rightarrow (\text{pro join } x)])$

wants $\forall x[\text{groupChat } x \rightarrow (\text{pro join } x)]$

LOWER

$\dfrac{\forall x[\text{groupChat } x \rightarrow []]}{\text{pro join } x}$

S

PRO$^\uparrow$ $\dfrac{\forall x[\text{groupChat } x \rightarrow []]}{\lambda y . y \text{ join } x}$

S

join$^\uparrow$ $\dfrac{\forall x[\text{groupChat } x \rightarrow []]}{x}$

every group chat

(56)   Daniele wants to join every group chat. $\forall > $ want

$\forall x[\text{groupChat } x \rightarrow \text{Dani wants}(\text{pro join } x)]$

LOWER

$\dfrac{\forall x[\text{groupChat } \rightarrow []]}{\text{Dani wants }(\text{pro join } x)}$

Daniele   ...

wants   ...

PRO to join every group chat

We can schematize what's going on here by boxing the point of the derivation at which LOWER applies:

- Daniele (wants $\boxed{\text{pro}^\uparrow \text{ S (join}^\uparrow \text{ S everyGroupChat)}}\downarrow$)

- $\boxed{\text{Daniele}^\uparrow \text{ S (wants}^\uparrow \text{ S (pro}^\uparrow \text{ S (join}^\uparrow \text{ S everyGroupChat)))}}\downarrow$

If you're more familiar with a treatment of scope-taking in terms of quantifier raising, then you can think of LOWER as being the correlate of the landing site of QR; semantic composition proceeds via S up until we encounter the landing site, at which point we switch back to "vanilla" semantic composition via A.

0.6.2   Scope rigidity

As we've seen however, when we're dealing with multiple scopally *mobile* expressions (such as quantifiers), continuation semantics derives surface scope readings by default. It is, therefore, well-suited to languages such as German and Japanese, which have been argued to display *scope-rigidity* (modulo semantic reconstruction amongst other potential exceptions).

The following example is from Kuroda (1970).

(57) a. *Dareka-ga   subete-no hon-o   yonda.*
          someone-NOM all-GEN   book-ACC read.
          "Someone read all the books." $\exists > \forall; \boldsymbol{\times} \forall > \exists$

      b. *Subete-no hon-o   dareka-ga   yonda.*
          all-GEN   book-ACC someone-NOM read.
          "Someone read all the books" $\forall > \exists; \exists > \forall$

There are also, of course, famous environments in English where we observe apparent scope-rigidity, such as the double-object construction (scope rigidity in the double-object construction is usually described as *scope freezing*).

(58) a. Daniele sent a syntactician every picture. $\exists > \forall; \boldsymbol{\times} \forall > \exists$
      b. Daniele sent a picture to every syntactician. $\exists > \forall; \forall > \exists$

Bobaljik & Wurmbrand (2012) posit a (violable) economy condition in order to express the preference for surface scope observed in many languages. Bobaljik & Wurmbrand make architectural assumptions that we aren't necessarily committed to here, but the spirit of Scope Transparency (SCoT) is very much in

line with a continuation semantics for quantifiers, where surface scope is the default, and inverse scope will only be achievable via additional application of the type-shifting rules posited.

(59) Scope Transparency (ScoT)
If the order of two elements at LF is $A > B$, then the order at PF is $A > B$.

We still need to account for the availability of *inverse scope readings* in English, and other languages however. We'll do this using *multi-story towers*.

0.7 Next week

Hopefully you're starting to get a feel for what continuations can accomplish with very few primitives. Next week, we'll pick up where we left off, with a discussion of:

- Inverse scope.

- Generalized con/dis-junction.

- *Indexed* continuations and a semantics for determiners.

- Continuations and exceptional scope

After that, Martin will take over with a re-assessment of QR.

Bibliography

Barker, Chris. 2002. Continuations and the Nature of Quantification. *Natural Language Semantics* 10(3). 211–242.

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and natural language* (Oxford studies in theoretical linguistics 53). Oxford University Press. 228 pp.

Beaver, David I. 2001. *Presupposition and assertion in dynamic semantics* (Studies in logic, language, and information). Stanford, California: CSLI. 314 pp.

Bobaljik, Jonathan David & Susi Wurmbrand. 2012. Word Order and Scope: Transparent Interfaces and the ¾ Signature. *Linguistic Inquiry* 43(3). 371–421.

Charlow, Simon. 2014. *On the semantics of exceptional scope*. Dissertation.

Danvy, Oliver & Andrzex Filinski. 1992. Representing Control: a Study of the CPS Transformation. *Mathematical Structures in Computer Science* 2(4). 361–391.

de Groote, Philippe. 2006. Towards a montagovian account of dynamics 16. 1–16.

Elliott, Patrick D. 2019a. *Fuck* compositionality. Slides from an invited talk at the DGfS workshop *Encoding emotive attitudes in non-truth-conditional meaning*.

Elliott, Patrick D. 2019b. Overt movement as higher-order structure building. unpublished manuscript. Leibniz-Zentrum Allgemeine Sprachwissenschaft.

Grove, Julian. 2019. *Scope-taking and presupposition satisfaction*. University of Chicago dissertation.

Gutzmann, Daniel. 2015. *Use-conditional meaning: studies in multidimensional semantics*. First edition (Oxford Studies in Semantics and Pragmatics 6). OCLC: ocn894201804. Oxford, United Kingdom: Oxford University Press. 304 pp.

Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*. 2011 edition - typesetting by Anders J. Schoubye and Ephraim Glick. University of Massachusetts - Amherst dissertation.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar* (Blackwell textbooks in linguistics 13). Malden, MA: Blackwell. 324 pp.

May, Robert. 1977. *The grammar of quantification*. Massachussetts Institute of Technology dissertation.

McCready, Elin. 2010. Varieties of conventional implicature. *Semantics and Pragmatics* 3(0). 8-1–57.

Partee, Barbara. 1986. Noun-phrase interpretation and type-shifting principles. In J. Groenendijk, D. de Jongh & M. Stokhof (eds.), *Studies in discourse representation theory and the theory of generalized quantifiers*, 115–143. Dordrecht: Foris.

Partee, Barbara & Mats Rooth. 1983. Generalized conjunction and type ambiguity. In *Meaning, use, and interpretation of language*, Reprint 2012, 361–383. Berlin, Boston: De Gruyter.

Potts, Christopher. 2005. *The logic of conventional implicatures* (Oxford Studies in Theoretical Linguistics 7). OCLC: ocm56964776. Oxford: Oxford University Press. 246 pp.

Shan, Chung-chieh. 2002. A continuation semantics for interrogatives that accounts for Baker's ambiguity. In Brendan Jackson (ed.), *Salt xii*, 246–265. Massachussetts Institute of Technology: Linguistic Society of America.

Shan, Chung-chieh. 2005. *Linguistic side effects*. Harvard University dissertation.

Shan, Chung-Chieh & Chris Barker. 2006. Explaining Crossover and Superiority as Left-to-right Evaluation. *Linguistics & Philosophy* 29(1). 91–134.

Wadler, Philip. 1994. Monads and composable continuations. *LISP and Symbolic Computation* 7(1). 39–55.

Bibliography

Barker, Chris. 2002. Continuations and the Nature of Quantification. *Natural Language Semantics* 10(3). 211–242.

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and natural language* (Oxford studies in theoretical linguistics 53). Oxford University Press. 228 pp.

Beaver, David I. 2001. *Presupposition and assertion in dynamic semantics* (Studies in logic, language, and information). Stanford, California: CSLI. 314 pp.

Bobaljik, Jonathan David & Susi Wurmbrand. 2012. Word Order and Scope: Transparent Interfaces and the ¾ Signature. *Linguistic Inquiry* 43(3). 371–421.

Charlow, Simon. 2014. *On the semantics of exceptional scope*. Dissertation.

Danvy, Oliver & Andrzex Filinski. 1992. Representing Control: a Study of the CPS Transformation. *Mathematical Structures in Computer Science* 2(4). 361–391.

de Groote, Philippe. 2006. Towards a montagovian account of dynamics 16. 1–16.

Elliott, Patrick D. 2019a. *Fuck* compositionality. Slides from an invited talk at the DGfS workshop *Encoding emotive attitudes in non-truth-conditional meaning*.

Elliott, Patrick D. 2019b. Overt movement as higher-order structure building. unpublished manuscript. Leibniz-Zentrum Allgemeine Sprachwissenschaft.

Grove, Julian. 2019. *Scope-taking and presupposition satisfaction*. University of Chicago dissertation.

Gutzmann, Daniel. 2015. *Use-conditional meaning: studies in multidimensional semantics*. First edition (Oxford Studies in Semantics and Pragmatics 6). OCLC: ocn894201804. Oxford, United Kingdom: Oxford University Press. 304 pp.

Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*. 2011 edition - typesetting by Anders J. Schoubye and Ephraim Glick. University of Massachusetts - Amherst dissertation.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar* (Blackwell textbooks in linguistics 13). Malden, MA: Blackwell. 324 pp.

May, Robert. 1977. *The grammar of quantification*. Massachussetts Institute of Technology dissertation.

McCready, Elin. 2010. Varieties of conventional implicature. *Semantics and Pragmatics* 3(0). 8-1–57.

Partee, Barbara. 1986. Noun-phrase interpretation and type-shifting principles. In J. Groenendijk, D. de Jongh & M. Stokhof (eds.), *Studies in discourse representation theory and the theory of generalized quantifiers*, 115–143. Dordrecht: Foris.

Partee, Barbara & Mats Rooth. 1983. Generalized conjunction and type ambiguity. In *Meaning, use, and interpretation of language*, Reprint 2012, 361–383. Berlin, Boston: De Gruyter.

Potts, Christopher. 2005. *The logic of conventional implicatures* (Oxford Studies in Theoretical Linguistics 7). OCLC: ocm56964776. Oxford: Oxford University Press. 246 pp.

Shan, Chung-chieh. 2002. A continuation semantics for interrogatives that accounts for Baker's ambiguity. In Brendan Jackson (ed.), *Salt xii*, 246–265. Massachussetts Institute of Technology: Linguistic Society of America.

Shan, Chung-chieh. 2005. *Linguistic side effects*. Harvard University dissertation.

Shan, Chung-Chieh & Chris Barker. 2006. Explaining Crossover and Superiority as Left-to-right Evaluation. *Linguistics & Philosophy* 29(1). 91–134.

Wadler, Philip. 1994. Monads and composable continuations. *LISP and Symbolic Computation* 7(1). 39–55.