Composing local semantic commitments

April 29, 2021

The dynamics of semantic commitments
semantic commitments, not local contexts

Abstract

Some theories of presupposition and of modality depend on identifying the *local context* of a subexpression: the net semantic commitments of the context of utterance after updating with the content of those subexpressions that have already been evaluated. Defining local contexts has involved either reasoning about the set of all possible syntactic completions, which is conceptually roundabout (not to mention impractical); or else stating separate update rules for each operator in parallel with their truth conditions, failing to capture the how local contexts depend only on truth conditions and order of evaluation. This note shows how to use continuations to compute local contexts directly from the composition of truth conditions. One reason this result is theoretically important is that it is fully compatible with a point-wise evaluation strategy, with no need to track the complete context set, in constrast with certain influential theories of presupposition and epistemic modality. On this view, sentences denote functions from a world to a truth value. The net result is a dynamic semantics in which the only thing that is tracked is the content of what has already been said.

Acknowledgments: Schlenker, Rothschild, Chung, Mandelkern. 2008.

1 Minimal dynamics

Something about natural language is dynamic. Order matters. Not only does context influence the evaluation of expressions, the evaluation of expressions influences the context with respect to which subsequent expressions get evaluated. For instance, evaluating an indefinite guarantees that a discourse referent will be available for subsequent anaphora (within certain limited domains). Some dynamic theories of meaning account for such facts by reconfiguring denotations as context update functions. But this gives far too much importance to the dynamic aspects of meaning. Other theories regulate dyanmic effects in parallel with the composition of meaning. But this denies that dynamic effects are properly semantic.

What, then, is the conceptually simplest possible dynamic semantics? The answer given here is one on which the scoreboard tracks exactly one thing: the semantic commitments of what has been said so far. Tracking commitments is implicitly necessary just in order to be able to compute truth conditions. The only question is how to make those commitments available to embedded expressions. I will show how to do this using a variation on standard continuation-passing-style techniques.

What made CCPs so attractive? They offer a simple way to track semantic commitments by tracking their net effect on the context set. Likewise, Schlenker's approach arrives at semantic commitments by incrementally tracking the syntactic form, and deriving the semantic commitments from that. The account here shows how to track semantic commitments directly, without either first cashing them out as context updates, or getting at them in a roundabout way through the syntactic forms that denote them.

[move to core technique] In a dynamic semantics, order matters. But the order of what? It is tempting to think that it is linear order, but that can't be right. Barker and Shan 2014 argue that the appropriate notion of order is *evaluation order*, the order in which expressions get evaluated. Their main arguments are based on a careful examination of weak crossover. This is not the place to relitigate that debate; instead, I will assume here that logical form encodes the relevant order relationships.

2 Local contexts and what they're good for

The two uses of local contexts I will emphasize here involve presupposition and epistemic modality. I'll discuss each in turn.

2.1 Presupposition satisfaction

Building on insights of Karttunen 1973 and Stalnaker 1973, Karttunen 1974 proposes an account of presupposition projection based on computing *local contexts*:

> In compound sentences, the initial context is incremented in a left-to-right fashion giving for each constituent a *local context* that must satisfy its presuppositions.

In particular, the local context for a right conjunct is the initial context updated with the content of the left conjunct.

(1) Ann has a cat, and Ann's cat is asleep.

(2) Ann's cat is asleep, and Ann has a cat.

In (1), the right conjunct presupposes that Ann has a cat. If κ is the intial context, then the local context for the second conjunct is κ+*Ann has a cat*. Clearly, this local context satisfies the presuppositions of the right conjunct for any choice of κ. In constrast, in (2), it is the left conjunct that presupposes that Ann has a cat. Its local context is the initial context, so the prediction is that (2) requires that κ satisfies the presupposition, which accounts for the impression that the right conjunct in (2) feels redundant.

Heim 1983 provides an influential implementation of Karttunen's strategy. She encodes the meaning of logical connectives as functions taking a complete context to an updated context. For instance, on Heim's account, a conjoined expression of the form "A and B" takes a context κ as input, and returns $(\kappa + A) + B$: the initial context updated with A, and the resulting intermediate context updated with B. This implementation makes it clear that the local context of the right conjunct is $\kappa + A$.

Although elegant and insightful, Heim's update semantics has been critized (notably by Soames 1989 and Schlenker 2007 et seq.) as being insufficiently explanatory. The complaint is that there are many update recipes for each logical connective that get the truth conditions right without delivering the appropriate local contexts. Because Heim defines the context updates on a per-connective basis, on Heim's account, Karttunen's overarching left-to-right pattern appears to be an accident rather than a principle.

In a series of papers, Schlenker (2007, 2008, 2009, 2010) develops an alternative theory on which local contexts are calculated in parallel with truth conditions. The local context for a given expression corresponds to the ... [fill in]. Deciding whether a given context satisfies a presupposition involves quantifying over all possible grammatical completions of the structure, and considering the logical form induced by each completion. Predictions equivalent to Heim's emerge in combination with special-purpose pragmatic principles (Be Articulate! and [fill in]).

This approach has some notable virtues. For one, it gives linear order an appropriately central and priviledged role: presupposition satisfaction is determined by the content of material that has already been pronounced, and not be material that is still to come. (Schlenker also provides a variant that is symmetric with respect to linear order, as I'll discuss below in section **??**.) In addition, this theory leaves semantic values as functions from evaluation coordinates (e.g., worlds and assignments) to extensions. In particular, logical connectives retain their traditional status as bivalent truth conditional operators.

However, the computation is somewhat roundabout, ranging from quantifying over possible syntactic completions to new pragmatic principles.

This paper shows how to reconstruct Schlenker's account in a purely semantic theory, working only with logical forms, and without relying on syntactic well-formedness or pragmatic principles. Local contexts are computed directly in the semantics, keeping the standard bivalent denotations for each of the logical connectives. The key techique involves a variation on what is known in the theory of computer programming languages as Continuation Passing Style (CPS). The relevance of these and related techniques to natural language semantics has been noted in Barker 2002, Shan 2001, de Groote 2001, and Barker and Shan 2014, among others. In section **??** I'll compare it in particular with the continuation-based reconstruction of Dynamic Predicate Logic in de Groote 2007.

I'll show how lifting a standard logical form into a continuation-passing style computation is provably equivalent to the original logical form denotation, yet simultaneously provides access to each expression's local context.

2.2 Epistemic modality

Epistemic modality provides independent motivation for considering a context update semantics. In the update semantics of Veltmann xxxx and Groenendijk et al. 1995, $\kappa+$ "Might p" is just κ if κ is consistent with p, and the empty context otherwise. This explains why the following sentence sounds contradictory:

(3) I won't win, but I might win.

On an epistemic interpreation of *might* (rather than a circumstantial or metaphysical interpretation), if the first conjunct is true, the second conjunct can't possibly be true. This falls out naturally on a context update view, since the local context against which the second conjunct gets evaluated will have already incorporated the content of the left conjunct, by hypothesis.

General theories of modality do not involve interrogating the evaluation context; rather, an accessibility relation (or, equivalently for our purposes, a conversational background characterizing a modal base) associates each evaluation point with a set of epistemically viable worlds. Mandelkern 2019 shows how to use local contexts to constrain appropriate accessibility relations: roughly, the set of epistimically accessible worlds must be a subset of the local context. He makes local contexts explicit by providing recursive rules in parallel with the truth conditions for each operator in the logical form, once again leavning the systematic patterns implicit.

One difference between the local contexts for presuppositions and for epistemic modals is that Mandelkern argues that local contexts for epistemics are symmetric rather than left to right. I will argue in favor of reconsidering that claim below. But whether or not the relevant conception of local contexts is left to right or symmetric, the same continuation-based techniques will provide a systematic account.

2.3 Anti-update

Heim's context change functions update entire contexts (sets of evaluation points). Likewise, Veltmann's update semantics crucially relies on providing epistemic modals with access to its entire local context at once. On the semantic reconstruction of local contexts here, expressions denote functions from evaluation points to extensions. In particular, on the fragment developed below, clauses denote functions from worlds and assignments to truth values. Nor is there any need to express constraints on epistemic modals in terms of contexts, rather than as constraints on invididual evaluation points. The claim, then, is that neither presupposition projection nor epistemic modality require tracking contexts. There is no doubt that discourse participants update their contexts in response in part to the content of utterances; however, that does not mean that context update is part of the semantics.

There is no doubt that the semantic value of an expression depends on context: the value of intensional expressions depend on the world of evaluation, and the value of anaphoric expressions depend on the operative assignment function. Following Lewis 1979, call the aspects of the context that are relevant for evaluation the 'scoreboard'. The essence of the dynamic semantics is the belief that evaluating some expressions not only depend on the score, but change certain values on the scoreboard.

What gets recorded on the scoreboard? Facts, worlds, sets of worlds? Or merely a record of what has been said so far?

3 Tracking semantic commitments

Continuations, monads, applicatives, functors. De Groote and anaphoric left context. The system. Examples.

The goal is to build a meaning incrementally in such a way that the content of expressions can be added in a order that is not fully determined by their hierarchical relationships. As recognized in the literature (Shan and Barker 2006, Schlenker 2008, Barker and Shan 2014), for obvious functional reasons the order of evaluation in most cases should be left to right, the same order in which we hear the expressions. There are situations, however, in which evaluation order diverges from simple linear order; quantifier scope relations are an example. In this note, I'll adopt the strategy of George, and assume that it is the job of Logical Form to encode any sort of evaluation order that is relevant for semantic interpretation.

3.1 The core technique

In order to track incremental semantic commitments, we're going to lift an ordinary logical form into a continuation-passing computation. For lexical items, this is achieved by applying the T combinator. This is simply an unrestricted version of Partee's 1987 LIFT type-shifter, which turns an individual-denoting expression into a generalized quantifier:

(4) $\textsc{t} = \lambda x\kappa.\kappa x$ e.g., $\textsc{t}\ \mathbf{ann} = \lambda P.P\ \mathbf{ann}$

This type shifter is a good candidate for being freely and generally available independently of the phenomena discussed here (see Barker 2020 for discussion).

For logical forms that involve function application, we apply the following novel combinator, which I will call H:

(5) $\textsc{h} = \lambda LRk.R(L(\lambda xy.k(xy)))$

As we will see, the net effect of inserting H is that the semantic contribution of the left argument (corresponding to the argument 'L') is evaluated immedidately, and the evaluation of the right argument ('R') is delayed, in anticipation of future specfication. As I'll explain below, H is a variant on standard continuation-passing style transforms; what is important at the moment is to simply observe how it enables computing incremental semantic commitments. Let's consider some simple examples.

(6) a. [Ann [[saw Bill] yesterday]]
b. $\textsc{h}\ (\textsc{t}\ \mathbf{ann})(\textsc{h}\ (\textsc{h}\ (\textsc{t}\ \mathbf{saw})\ (\textsc{t}\ \mathbf{bill}))(\textsc{t}\ \mathbf{yesterday})))$
c. $\rightsquigarrow_\beta \lambda k.k(\mathbf{yesterday}\ (\mathbf{saw}\ \mathbf{bill})\ \mathbf{ann})$

The expression in (6c) is the lifted logical form in (6b) after beta-reduction. It is easy to recover the pre-lifted value of the sentence by simply applying (6c) to $\textsc{i} = \lambda x.x$, the identity combinator. When $\textsc{i}$ is used in this role, it is known as 'the trivial continuation'.

We're now ready to consider some simple examples involving presuppositions. Let's start with the key minimal pair that motivates Schlenker's approach.

(7) It rained, and Ann knows it rained.

(8) Ann knows it rained, and it rained.

In (7), the content of the left conjunct entails the presupposition triggered by the verb *know*, namely, that its complement is true. As a result, the fact that it rained is asserted, and the sentence as a whole does not presuppose anything. In contrast, in (8), the information in the right conjunct arrives too late to satisfy the presupposition triggered in the left conjunct, and the sentence as a whole presupposes that it rained, which is why the right conjunct sounds redundant.

Here's the lifted version of the first element of the minimal pair:

(9) a. [It rained [and [Ann [knows it rained]]]]
b. $\textsc{h}\ (\textsc{t}\ \mathbf{rain})(\textsc{h}\ (\textsc{t}\ \mathbf{and})(\textsc{h}\ (\textsc{t}\ \mathbf{ann})(\textsc{h}\ (\textsc{t}\ \mathbf{knows\text{-}rain}))))$
c. $\rightsquigarrow_\beta \lambda\kappa.\textsc{t}\ \mathbf{knows\text{-}rain}\ (\lambda P.\kappa(\mathbf{and}\ (P\ \mathbf{ann})\ \mathbf{rain}))$

After application to the trivial continuation, the argument to the (lifted) presupposition-triggering verb phrase *knows it rained* is $\lambda P.(\mathbf{and}\ (P\ \mathbf{ann})\ \mathbf{rain})$: a function from verb phrase meanings P to the proposition that P applies to Ann, and it rained. More precisely, no matter what value P takes, this function applied to P will entail that it rained. Thus the argument contains all of the information necessary to conclude that it has rained.

Here is the second element of the minimal pair for comparison:

(10) a. [[Ann [knows it rained]] [and [it rained]]]
b. $\textsc{h}\ (\textsc{h}\ (\textsc{t}\ \mathbf{ann})(\textsc{t}\ \mathbf{knows\text{-}rain}))(\textsc{h}\ (\textsc{t}\ \mathbf{and})(\textsc{t}\ \mathbf{rain}))$
c. $\rightsquigarrow_\beta \lambda\kappa.\textsc{t}\ \mathbf{knows\text{-}rain}\ (\lambda P f.\kappa(f(P\ \mathbf{ann})))\ (\mathbf{and}\ \mathbf{rain})$

In this case, after application to the trivial continuation, the argument to the (lifted) verb phrase *knows it rained* is $(\lambda P f.f(P\ \mathbf{ann}))$: a function from a verb phrase meaning P and a one-place truth-value operator f to $f(P\ \mathbf{ann})$. The function f will turn out to be $\lambda p.\mathbf{and\ rain}\ p$, but that information does not appear in the first argument to (the lifted) *knows it rained*. The (lifted) verb phrase's argument does not guarantee that it rained, so the verb phrase meaning has no choice but to search for satisfaction of its presupposition in the background situation.

The idea here is that the semantic argument to each expression represents its local context: the semantic content of all and only the material to the left. The argument will contain variables that serve as placeholders for the denotations of expressions to the right, but those placeholders will contain no lexical content. As a result, each expression will have direct semantic access to the semantic commitments imposed by its left context.

Some important points to note right off the bat:

- The lifted computation is purely semantic, and, unlike Schlenker 2009, 2010 does not involve quantifying over any class of syntactic completions.
- The logical connective *and* receives no special treatment here. It bears its standard bivalent truth conditional meaning (for now; see below), and it undergoes the same simple lifting operation as any other lexical item.
- There is no reference to the context set, or any set of worlds
- The left-right asymmetry is systematic across all expressions. Furthermore, the asymmetry is located in a single place in the system, namely, in the H combinator. (More on this below.)

This is progress towards fulfilling the intoxicating promise in Heim 1983: after lifting, merely stating the truth conditions of the expressions is sufficient to determine their behavior with respect to presupposition projection.

4 Managing presuppositions

At this point, we already have everything we need to reconstruct Schlenker's account of presupposition satisfaction in a purely semantic theory. Instead of reasoning about possible syntactic completions (and their denotations), we will directly construct the corresponding semantic functions.

The only slight complication arises because local contexts sometimes have unsaturated arguments corresponding to as yet unseen rightward expression, as we saw above in (10). This means we need to find a generalized notion of entailment that operates on unsaturated functions. Say that a function f *guarantees* a proposition p ('$f \Rightarrow p$') just in case every way of saturating the arguments of f leads to a proposition that entails p.

$$f \Rightarrow p \text{ iff } f \text{ has type } \mathtt{t} \text{ and } f \rightarrow p, \tag{11}$$

$$\text{or } f \text{ has type } a \rightarrow b \text{ and } \forall x \text{ with type } a, fx \Rightarrow p \tag{12}$$

For instance, $\lambda p.$**and rain** p guarantees **rain**, since no matter how we instantiate the propositional argument p, the result (namely, **and rain** p) will entail **rain**. This notion of a guarantee is just a semanticization of Schlenker's metalinguistic reasoning about all possible completions.

We can now consider the following theory of presupposition satisfaction:

(13) **Presupposition satisfaction**: the presupposition p of an expression with local context κ is satisfied just in case κ guarantees p, that is, just in case $\kappa \Rightarrow p$.

Normally, presuppositions can be satisfied either by the part of their local context provided entirely by the content of the sentence in which they occur, or else by some information present in the utterance context. This will fall out automatically, as we'll see in a moment.

Up to this point, lexical items have always entered the computation after lifting with T. But nothing prevents us from writing a special lexical entry that could participate in the lifted computation directly. This is exactly how a lexical item will declare a presupposition:

(14) know: $\lambda \kappa p : \kappa \Rightarrow p.\kappa\,(\mathbf{know}\ p)$

Using the Heim and Kratzer notation for specifying presuppositions, this lexical entry says that the verb phrase *know p* is defined only if its local context guarantees p, in which case the value of the verb phrase is the local context κ applied to the function **know**.

If an expression contains more than one presupposition trigger, leftward triggers must be satisfied before the presuppositions of the rightward triggers are considered. All presuppositions must be satisfied in order for the denotaiton of the expression as a whole to be well defined.

Schlenker 2008 proves that his account can reproduce the predictions of Heim's 1983 theory concerning presuppositions exactly. Because of the deliberately close correspondence between Schlenker's principles and the construction here, the theory here is equally able to reconstruct Heim's theory, if desired.

5 Bounded modality

Wittgenstein noticed that the truth conditions of epistemic modals interact with nearby expressions.

(15) It's not raining, and it might be raining.

In (17), the truth of the first conjunct guarantees that the second conjunct will be false: if it's not raining, then it is not epistemically possible that it is raining. Update semantics such as Groenendijk et al. 1995 provide an elegant explanation: the local context with respect to which the right conjunct gets evaluated will have already been updated with the information that it is not raining. We need only assuming that epistemic modals have semantic access to their local context. In GSV, this is direct, since like all context update functions, epistemic clauses operate directly on their local context. But this strategy does not generalize to other modalities. Mandelkern 2020 suggests the following strategy: "The basic idea is that epistemic modals are quantifiers over accessible worlds, as the standard theory has it; but, crucially, their domain of quantification is limited by their local contexts." In particular, the set of epistemically accessible worlds is required to be a subset of the local context. Mandelkern calls this "bounded modality".

It is easy to implement bounded modality here.

(16) might = $\lambda p \lambda \kappa \lambda w. \exists w' \in \text{DOX } w : \kappa w' \wedge p w'$

Here, κ is the local context, and *might p* will be true just in case there is an epistemically accessible world w' that satisfies the local context and that makes p true.

The status of (**??**) with the conjuncts reversed is different in an important way.

(17) It might be raining, and it's not raining.

As in GSV's analysis, (17) is consistent, but not coherent: an evaluation point will survive update if corresponds to a world in which it not raining, but from which there is an epistemically accessible world in which it is raining. It is incoherent as an assertion, however, since if our epistemic situation is such that we don't know whether its raining, we're in no position to assert a sentence that entails that it's raining.

(18) Either it might be raining and it is raining, or it might be raining and it's not raining.

(19) If it might be raining, it's not raining.

False, or contradictory, or capable of being true? If true, bounding modality is not really symmetric.

My opinion is that a left-to-right theory of bounded modality is viable. But if you prefer a symmetric theory like the one that Mandelkern advocates, the symmetric local contexts in the next section will give the Mandelkern analysis.

6 Symmetrical continuations

7 Consequences

So far all we have done is re-engineer Schlenker's theory of presupposition. But this has two important consequences...

8 Details

So our starting point will be a Logical Form in which quantifier scope relations have already been disambiguated. It would be possible to arrive at the desired meanings directly, by modifying the interpretation rules for Logical Forms. However, it will be helpful conceptually to present the scheme as a transform, in the style of continuation-passing transforms from the computer science literature (Plotkin 1975, Danvy and co. 200X). Following the conventions in Heim and Kratzer 1998 the standard interpretation function for Logical Forms is quite simple, and depends on the types of the subexpressions.

Informal first, then the full treatment?

It will be helpful to define some standard combinators (SKITC), and one non-standard combinator: L, for "left to right incremental composition".

```
S = \x y z.x a (y z)
K = \x y.x
I = \x.x
T = \x y.y x
C = \x y z.x z y
L = \f l r k.r(l(\xy.k(fxy)))
```

```
Logic Form element            transform                    gloss

[c]                           c                            constant
[i]                           x_i                          index
[t_i]                         x_i                          trace
[gamma_{B->A} beta_B]         L I [gamma] [beta]           fn application
[beta_B gamma_{B->A}]         L T [beta] [gamma]           backwards fn application
[i (gamma_{B->A} beta_B)]     L S [i gamma] [i beta]       predicate abstraction
[i (beta_B gamma_{B->A})]     L (C S) [i beta] [i gamma] predicate abstraction
[i t_i]                       T I
[i *]                         T (K *)

Example: [everyone [1 [ann [saw t_1]]]] => (everyone (\lambda x_1. saw x_1 ann))
```

Continuation monad:

```
M a = (a -> b) -> b
eta x = \k.kx
m * f = \k.m(\a.fak)
```

9 Attitudes

10 Symmetry

stating presuppositions, bounds on modality

attitudes: imposition of local context

11 monads, functors, and all that

definites in MM's system are a problem, since they need to quantify over the entire context