Composing local contexts

Chris Barker, *NYU Linguistics*

February 3, 2022

Final draft (preprint), to appear in *Journal of Semantics*

Abstract

An expression's presuppositions must be satisfied by its *local context*, that is, by the utterance context updated with the content of expressions that have already been evaluated. Traditional dynamic approaches track local context by remaking clause denotations into context update functions. This requires crafting for each semantic operator an update recipe that is not fully determined by its truth conditions, failing to capture how local contexts depend only on truth conditions and order of evaluation. In other theories, computing local contexts involves reasoning about the set of all possible grammatical syntactic completions, which relocates local contexts outside of the semantics. I show how to build local contexts systematically and uniformly as part of the composition of ordinary truth conditions. The result is a minimalist dynamic semantics in which the only thing that is dynamically tracked is the semantic content of what has already been said.

Acknowledgments: Thanks to Amir Anvari, WooJin Chung, Simon Goldstein, Matthew Mandelkern, Philippe Schlenker, and Yasutada Sudo. Historical note: in 2008, I circulated a brief technical note that included the basic compositional idea, which I made available on `semanticsarchive.net` in 2011. That is the work cited in Schlenker 2010b as Barker 2008.

1 Towards a minimalist dynamics

Something about natural language is dynamic: not only does context partially determine the value of expressions, the evaluation of expressions partially determines the context with respect to which subsequent expressions get evaluated.

Some dynamic theories account for dynamic effects by reconceiving clause denotations as context update functions, and then deriving truth conditions from the update recipe. But this gives too much importance to the dynamic aspects of meaning—dynamic effects are the tail and truth conditions are the dog, not the other way around. Other theories regulate dynamic effects purely in the pragmatics, computed separately but in lock step with the composition of meaning. This denies that dynamic effects are properly semantic, despite the fact that the constraints imposed by local contexts are purely a matter of semantic entailment.

What, then, is the conceptually simplest possible dynamic semantics? The answer given here is that the compositional semantics tracks exactly one thing: the semantic content of what has been said so far. After all, some form of content tracking is necessary just in order to be able to compute truth conditions. The only question is how to make that incremental record explicitly available to embedded expressions. I will show how to do this using a novel variation on standard continuation-passing-style techniques, borrowing from the theory of programming languages. On the continuation approach, dynamic effects depend only on truth conditions and the order in which expressions are evaluated. The net result is a truly minimal dynamics semantics in which the only thing that is dynamically tracked is the semantic content of what has already been said.

2 Local contexts and what they're good for

There are three traditional empirical motivations for going dynamic: presupposition satisfaction, epistemic modality, and donkey anaphora. I will discuss the first two below, but a detailed discussion of anaphora will have to wait for another occasion.

This section is theoretical background and empirical motivation; nothing here is crucial for following the presentation of the positive view, so you're welcome to skip to section 3 if you know the literature and you're impatient to see how the technique works.

2.1 Local contexts constrain presupposition satisfaction

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

(1) 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, in a conjoined sentence, the local context for a right conjunct is the initial context updated with the content of the left conjunct.

(2) It's raining, and Ann knows it's raining.

(3) Ann knows it's raining, and it's raining.

In (2), the right conjunct presupposes that it is raining. Thanks to the presence of the left conjunct, however, the local context for the second conjunct is $\kappa +$ *It's raining*, where κ is the initial context. Clearly, this local context satisfies the presuppositions of the right conjunct for any value of κ. In contrast, in (3), it is the left conjunct that presupposes that it is raining. Nothing has been evaluated yet, so its local context is the initial context. The prediction is that the initial context must satisfy the presupposition, which accounts for the impression that the right conjunct in (3) feels redundant.

Heim 1983 provides an influential implementation of Karttunen's strategy. She encodes the meaning of clauses as context update functions, that is, as functions from a context to an updated context. For instance, 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 analysis makes it clear that the local context of the right conjunct is $\kappa + A$.

Although elegant and insightful, Heim's update semantics has been criticized, notably by Soames 1989:597 and Schlenker 2007 et seq., as being insufficiently explanatory. The complaint is that there exist update recipes for the logical connectives that get truth conditions right without delivering the appropriate local contexts. Because Heim must therefore define the context updates on a per-connective basis, the overarching left-to-right pattern identified by Karttunen appears to be an accident rather than a principle.

George 2014 shows how to impose explanatory regularity in a traditional (non-dynamic) system. They do this by adopting a trivalent logic, and then providing systematic rules for lifting ordinary bivalent meanings into the order-asymmetric trivalent logic. I am sympathetic to George's explanatory goals, and, like George, the main technique developed here will involve lifting ordinary logical forms into an order-sensitive computation. Unlike George's account, logical connectives here are not special here in any way, and there is no need to resort to a trivalent logic: the logical operators remain bivalent even after being lifted into the continuation fragment, and the lifting operation applies uniformly to expressions of any type.

In part reacting to the perceived shortcomings of the dynamic update program, Schlenker (2007, 2008a, 2008b, 2008c, 2009, 2010a, 2010b) develops a theory on which local contexts are purely pragmatic, and calculated in parallel with truth conditions. Given an incomplete utterance, a local context is computed by quantifying over all possible grammatical syntactic completions, and then considering certain semantic denotations related to each completion.

The research reported here owes an obvious debt to Schlenker's work, which has a number of notable virtues. For one, it gives linear order an appropriately central and privileged role. In addition, it leaves semantic values as traditional functions from evaluation points to extensions, rather than replacing them with context update functions. In particular, logical connectives retain their traditional status as bivalent truth conditional operators.

However, by quantifying over possible syntactic completions, this theory locates the computation of local semantic contexts outside of the semantics. Yet the bottom line is always purely semantic. That is, whether a presupposition is locally satisfied ultimately depends only on the semantic content of the expressions that have already been evaluated. The grammaticality of possible syntactic completions play no essential role.

As Schlenker 2007:328 puts it, on his system, "the only information that needs to be updated concerns the words that the speech act participants have pronounced." On the account here, the only information that needs to be updated concerns the *semantic content* of the expressions that have already been evaluated.

2.2 Local contexts constrain epistemic modality

Might p can be true even when *p* happens to be false. Yet, as Wittgenstein 1958[1953]:192 noticed, there is something wrong about asserting *might p* and denying *p* in the same breath:

(4) It isn't raining, but it might be raining.

(5) It might be raining, but it isn't raining.

Update semantics such as Groenendijk et al. 1996 and Veltman 1996 provide an elegant explanation for (4): assuming that the truth of *might p* depends on the local context, the local context of the right conjunct will guarantee it's not raining, so by the time the right conjunct is evaluated, it is no longer true. We can think of this kind of analysis as a dynamic contradiction.

The status of (5) is different. On the update analysis, which assumes a left to right evaluation order, (5) is not a dynamic contradiction—there are initial contexts that will update to a non-null output context. However, it is defective from the point of view of the norms of assertion. The reason is that any epistemic state that would justify asserting *It might be raining* would be one in which rain is a live possibility, which means that that epistemic state would not justify asserting a sentence that entails that it's not raining.

One challenge for the update view is that it handles epistemic modality in a way that is quite different from the standard treatment of modality. On the standard treatment, the truth conditions of all modals, including epistemic modals, are expressed in terms of an accessibility relation (or, equivalently for our purposes, in terms of a conversational background that characterizes a modal base). Mandelkern 2019 offers an approach that reconciles the dynamic explanation with the standard view on modality. On that account, "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." Mandelkern calls this "bounded modality".

The account of epistemic modality is one of the victories of dynamic update semantics, since the account follows from the same update semantics that were motivated by the problem of local presupposition satisfaction. We'll see that the same technique proposed here for composing local contexts in order to account for presupposition satisfaction can account for the dynamics of epistemic modality as well.

With these two empirical targets in view, we can turn to the proposal.

3 Composing local contexts

In a dynamic semantics, order matters. But the order of what? Following Karttunen and Schlenker, it's tempting to say plain linear order. In Shan and Barker 2006 and Barker and Shan 2014, we say that it's *evaluation* order, which differs from simple linear order at least with respect to quantifier scope relations (see also Chung in prep for relevant discussion). For our purposes here, I'll assume that whatever the correct conception of order turns out to be, it is part of the job of Logical Form to encode that. So, just like George 2014, we'll take LFs as our starting point. (Though see also the discussion of linear order in section 7.5.)

3.1 Continuizing a Logical Form

In order to track incremental semantic composition, we're going to lift an ordinary logical form into a continuation-passing style computation.[1] So if α is an ordinary Logical Form, I'll write $\boxed{\alpha}$ for its continuized version, as defined immediately below. What makes continuation-passing style useful here, as we will see, is that it provides explicit semantic access to incremental semantic context. I'll say more about these conceptual underpinnings below in section 6, but it will be helpful to see the approach in action first.

Logical Forms contain three main types of configurations: lexical items, function application, and predicate abstraction. The first two are discussed immediately below. There is nothing problematic about predicate abstraction, but we won't need it to illustrate how the system works, so the relevant details appear in a brief appendix.

Lexical items: for a lexical item α that does not trigger any presuppositions, $\boxed{\alpha}$, the continuized version of α, is simple:

(6) $\boxed{\alpha} = \lambda\kappa.\kappa\alpha$ Continuized lexical item

Here and throughout, I'll use the variable κ to stand for the local context of the expression in question. So the continuized version of α takes its local context and applies it to the ordinary (pre-continuized) value of α.

Because the semantic type of lexical items varies, the type of κ will also vary; that is, the continuization operation schematized in (6) is polymorphic. As many readers will have already noticed, this operation is just a generalized version of Partee's 1987 LIFT type-shifter. Partee's type-shifter turns an individual-denoting expression into a generalized quantifier. For example, if **ann** is an individual

[1]In addition to my own earlier work on continuations, including Barker 2002 and Shan and Barker 2006, as summarized in Barker and Shan 2014, the idea of using continuations to track local context was inspired in part by de Groote's 2006 continuation-based dynamic grammar. See de Groote and Lebedeva 2010 and Lebedeva 2012 for a different approach using continuations to account for presupposition accommodation.

constant of type $\mathtt{e}$, then $\textsc{lift}(\mathbf{ann}) = \lambda\kappa.\kappa\,\mathbf{ann} = \boxed{\mathbf{ann}}$, a generalized quantifier of type $(\mathtt{e} \to \mathtt{t}) \to \mathtt{t}$.

Function application: as we'll see in section 7.3, there is more than one way to continuize function application. The variant we need at this point delivers local contexts that include only previously evaluated content. So, given an instance of function application $[\beta\ \gamma]$ in which β and γ are ordinary Logical Forms, $\boxed{[\beta\ \gamma]}$, its continuized version, is given by:

(7) $\boxed{[\beta\ \gamma]} = \lambda\kappa.\boxed{\gamma}(\boxed{\beta}(\lambda xy.\kappa[x\,y]))$ Continuized function application

This continuization schema is the heart of the proposal in this paper. On the right hand side of the equation, '$\boxed{\beta}$' is the continuized version of the sub-LF β, and '$\boxed{\gamma}$' is the continuized version of the sub-LF γ.

It's important to note that the logical-form brackets in (7) correspond to Heim-and-Kratzer 1998 style type-driven interpretation. That is, if x has type $A \to B$ and y has type A, then $[x\,y] = x(y)$, and has type B. But if x has type A and y has type $A \to B$, then $[x\,y] = y(x)$, which once again has type B. In other words, as usual, function application in logical form either works left to right or right to left, whichever is compatible with the types of the elements. This is crucial to allow logical forms to accurately reflect the order of evaluation in natural langauge, given that arguments sometimes occur to the left and sometimes to the right of the predicates they combine with.

Continuization does not add or subtract information from the Logical Form. Rather, it merely puts the pieces together in a way that makes incremental semantic context available to semantic denotations. In fact, the net result of the continuized computation is (almost!) identical to the original LF. To see this, here is a simple example involving a single instance of function application:

(8) Ann laughed.

$$
\begin{aligned}
&\boxed{[\textbf{ann laughed}]}\\
&= \lambda\kappa.\boxed{\textbf{laughed}}(\boxed{\textbf{ann}}(\lambda xy.\kappa[x\,y])) && \text{(continuized fn application, (7))}\\
&= \lambda\kappa.\boxed{\textbf{laughed}}((\lambda\kappa'.\kappa'\,\textbf{ann})(\lambda xy.\kappa[x\,y])) && \text{(cont'd lexical item, (6))}\\
&= \lambda\kappa.\boxed{\textbf{laughed}}((\lambda xy.\kappa[x\,y])\,\textbf{ann}) && \text{(beta reduction)}\\
&= \lambda\kappa.\boxed{\textbf{laughed}}(\lambda y.\kappa[\textbf{ann}\,y]) && \text{(beta reduction)}\\
&= \lambda\kappa.(\lambda\kappa'.\kappa'\,\textbf{laughed})(\lambda y.\kappa[\textbf{ann}\,y]) && \text{(cont'd lexical item, (6))}\\
&= \lambda\kappa.(\lambda y.\kappa[\textbf{ann}\,y])\,\textbf{laughed} && \text{(beta reduction)}\\
&= \lambda\kappa.\kappa[\textbf{ann laughed}] && \text{(beta reduction)}
\end{aligned}
$$

The only difference between the value of the continuized logical form and the original (uncontinuized) logical form is the presence of κ in the final line. This κ represents the semantic context within which the expression *Ann laughed* is being evaluated. In order to recover the standard truth conditions of the sentence, simply apply the continuized result to the identity function $\text{I} = \lambda\kappa.\kappa$ (i.e., the empty context), like this: $(\lambda\kappa.\kappa[\textbf{ann laughed}])\,\text{I} = \text{I}\,[\textbf{ann laughed}] = [\textbf{ann laughed}]$.

So continuized logical forms encode the exact same denotation as the uncontinuized logical form, up to application to the identity function. In Plotkin's 1975 paper introducing a number of continuation passing style transforms, he proves a theorem he calls Simulation: for any expression α, $\boxed{\alpha}\,\text{I} = \alpha$ (where '$\boxed{\alpha}$' is any of several continuization transforms defined by Plotkin). Simulation holds for the continuization scheme here. To see why, consider a lemma I'll call Flip: for any logical form α, $\boxed{\alpha} = \lambda\kappa.\kappa\alpha$. Flip clearly holds for lexical items. For the inductive case, assume Flip holds for for $\boxed{\beta}$ and $\boxed{\gamma}$. Then we can follow the reasoning in the derivation of *Ann laughed* immediately above exactly, replacing $\boxed{\textbf{ann}}$ with $\boxed{\beta}$ and replacing $\boxed{\textbf{laughed}}$ with $\boxed{\gamma}$. Given Flip, Simulation follows immediately. (See the appendix for a proof that Flip and Simulation hold in the presence of predicate abstraction.)

In other words, continuization is just a different way of arranging the same compositional elements, and does not itself change the semantic value of a logical form.

3.2 Defining local contexts

Here is the core idea of using continuations to characterize local contexts:

(9) **Local semantic context**. The first argument of a continuized expression represents its local semantic context.

Given this convention, inspecting the continuization schema in (7) reveals how it builds local contexts that contain only previously-evaluated expressions. Focusing first on $\boxed{\gamma}$, its first semantic argument is $\boxed{\beta}(\lambda xy.\kappa[xy])$. Since this expression includes $\boxed{\beta}$, it is clear that the local context of the right hand element includes the content of the left hand element. Next, focusing on $\boxed{\beta}$, its first semantic argument is $\lambda xy.\kappa[x\,y]$, which includes only κ, that is, the initial context within which the function application as a whole is being evaluated. So expressions on the right are evaluated in the context of expressions that precede them, but expressions on the left are evaluated without access to the content of the expressions that follow them.

In order to see how this plays out in a concrete example, it will be helpful to work through the minimal pair discussed above in (2) and (3), showing the key steps in the evaluation of the continuized expression. The goal of the computation in each case is to identify the local context for the verb phrase *knows it rained*. As the evaluation proceeds, because of the way that the continuization schema in (7) reverses the original order of logical form elements, progressively more deeply embedded elements will swap order with each other, somewhat like braiding hair.

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

$$\begin{aligned}
&\boxed{[\textbf{rain}\ [\textbf{and}\ [\textbf{ann know-rain}]]]} \\
=&\lambda\kappa.\boxed{[\textbf{and}[\textbf{ann know-rain}]]}(\lambda y.\kappa[\textbf{rain}\ y]) && (7, 6, \beta, \beta) \\
=&\lambda\kappa.\big(\lambda\kappa'.\boxed{[\textbf{ann know-rain}]}(\lambda y'.\kappa'[\textbf{and}\ y'])\big)(\lambda y.\kappa[\textbf{rain}\ y]) && (7, 6, \beta, \beta) \\
=&\lambda\kappa.\boxed{[\textbf{ann know-rain}]}(\lambda y'.\kappa[\textbf{rain}\ [\textbf{and}\ y']]) && (\beta, \beta) \\
=&\lambda\kappa.\big(\lambda\kappa'.\boxed{\textbf{know-rain}}(\lambda y.\kappa'[\textbf{ann}\ y])\big)(\lambda y'.\kappa[\textbf{rain}\ [\textbf{and}\ y']]) && (7, 6, \beta, \beta) \\
=&\lambda\kappa.\boxed{\textbf{know-rain}}(\lambda y.\kappa[\textbf{rain}\ [\textbf{and}\ [\textbf{ann}\ y]]]) && (\beta, \beta)
\end{aligned}$$

The verb phrase *knows it rained*, of course, presupposes that it rained. Assume for now that $\boxed{\textbf{knows-rained}}$ is the continuation-level presupposition-carrying denotation of the verb phrase, whatever that turns out to be; section 4.3 will explain in detail how to compute an appropriate denotation compositionally.

Given that a continuized expression's local context is its first semantic argument, the local semantic context for the verb phrase *knows it rained* according to (10) is $\lambda y.\kappa[\textbf{rained}\ [\textbf{and}\ [\textbf{ann}\ y]]]$: a function mapping any verb phrase meaning y to the result of updating the initial context κ with the proposition that it rained, and that y applies to Ann. Reasoning now in the manner of Schlenker 2009, it is clear that no matter what value y takes, and no matter what information is present in the intitial context κ, the proposition $\lambda y.\kappa[\textbf{rained}\ [\textbf{and}\ [\textbf{ann}\ y]]]$ will entail that it rained. (Sections 4 and 5 will make this reasoning precise.) In other words, the local semantic context for the verb phrase contains enough information to guarantee that the presupposition in question has been locally satisfied.

Turning now to (3), the second element of the minimal pair:

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

$$
\begin{aligned}
&\boxed{[[\textbf{ann know-rain}]\ [\textbf{and rain}]]} \\
=&\lambda\kappa.\boxed{[\textbf{and rain}]}\Big(\boxed{[\textbf{ann know-rain}]}(\lambda xy.\kappa[x\ y])\Big) && (7)\\
=&\lambda\kappa.\big(\lambda\kappa'.\kappa'[\textbf{and rain}]\big)\Big(\boxed{[\textbf{ann know-rain}]}(\lambda xy.\kappa[x\ y])\Big) && \text{(Flip)}\\
=&\lambda\kappa.\boxed{[\textbf{ann know-rain}]}(\lambda xy.\kappa[x\ y])[\textbf{and rain}] && (\beta)\\
=&\lambda\kappa.\big(\lambda\kappa'.\boxed{\textbf{know-rain}}(\lambda y'.\kappa'[\textbf{ann}\ y'])\big)(\lambda xy.\kappa[x\ y])[\textbf{and rain}] && (7, 6, \beta, \beta)\\
=&\lambda\kappa.\boxed{\textbf{know-rain}}\ (\lambda y'y.\kappa[[\textbf{ann}\ y']\ y])\ [\textbf{and rain}] && (\beta, \beta)
\end{aligned}
$$

In this case, the local context of the verb phrase *knows it rained* is $\lambda y'y.\kappa[[\textbf{ann}\ y']\ y]$: a function from any verb phrase meaning y' and any one-place truth-value operator y to the proposition that $\kappa[[\textbf{ann}\ y']\ y]$. Eventually, as processing continues, y' will be replaced with the at-issue component of the verb phrase *knows it rained*, and y will be replaced by the function on truth values denoted by *and it rained*. But at the point at which $\boxed{\textbf{knows-rained}}$ combines with its first semantic argument, that argument is simply $\lambda y'y.\kappa[[\textbf{ann}\ y']\ y]$. The only thing that this local context guarantees is that the verb phrase will be predicated of Ann. In particular, there is no information about whether it is or isn't raining. As long as we assume that the verb phrase must decide whether its presuppositions are satisfied based only on its local semantic context, the continuization strategy correctly predicts that the presupposition triggered by the verb can only be satisfied in (11) if it is already entailed by the initial context κ.

So the hypothesis that the first semantic argument to a continuized expression represents its local context accounts for the contrast in (2) and (3).

Some important points to note right off the bat:

- The continuized computation is purely semantic, and, unlike the pragmatic account in Schlenker 2007 et seq., 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, and it undergoes the same simple lifting operation as any other lexical item.
- The left-right asymmetry is systematic across all LF expressions. Furthermore, the asymmetry is located in a single place in the system, namely, in the rule for continuized function application given in (7) (see discussion in section 7.3).

This is progress towards fulfilling the intoxicating promise of a dynamic semantics: we now have a way of dynamically composing local contexts in the semantics that depends only on truth conditions and order of evaluation.

4 Presupposition satisfaction

4.1 Semantic commitments

In order to make predictions about presupposition satisfaction, we need to associate each local context with a suitable set of entailments. The main complication with doing this is that our local contexts can have unsaturated arguments corresponding to as yet unevaluated expressions, as we saw above for the local context computed in (10), namely, $\lambda y.\kappa[\mathbf{rained}\ [\mathbf{and}\ [\mathbf{ann}\ y]]]$ (recall that κ here represents the initial context for the relevant utterance of (10)). What should count as the relevant entailments of an arbitrary local context f?

The obvious answer (and the one that I will pursue here) is that the relevant entailments associated with f are the entailments that are present no matter how the arguments of f are instantiated.

That sounds straightforward enough. However, there are two complementary perspectives on what this could mean. First, we can consider the situations that are ruled out no matter how the arguments of f are instantiated. For instance, if $f = \lambda p.\textbf{rain} \wedge p$, then fp will entail that it rained no matter how p is instantiated. So **rain** is an upper bound on the set of situations compatible with the information in f: non-rain situations are excluded.

The second perspective is dual to the first: we can also consider what situations are ruled *in* no matter how the arguments of f are instantiated. For instance, if $f = \lambda p.\textbf{rain} \vee p$, rain worlds will entail fp no matter how p is instantiated. So now **rain** is a lower bound on the set of situations compatible with the information in f: rain situations are included.

We can define the least upper bound of f's semantic commitments ('$\lceil f \rceil$') and the greatest lower bound ('$\lfloor f \rfloor$') as follows:

$$\lceil f \rceil = \begin{cases} f & \text{if } f \text{ has type } \texttt{st} \\ \exists x_{\texttt{a}}.\lceil fx \rceil & \text{if } f \text{ has type } \texttt{a} \to \texttt{b} \end{cases} \tag{12}$$

$$\lfloor f \rfloor = \begin{cases} f & \text{if } f \text{ has type } \texttt{st} \\ \forall x_{\texttt{a}}.\lfloor fx \rfloor & \text{if } f \text{ has type } \texttt{a} \to \texttt{b} \end{cases} \tag{13}$$

These definitions will be well-defined for any discourse whose final semantic result is a set of worlds of type st.[2] So $\lceil f \rceil$ is the most informative (smallest) proposition that is entailed by each way of saturating the arguments of f. Dually, $\lfloor f \rfloor$ is the least informative (largest) proposition that entails each way of saturating the arguments of f.

Accounting for presupposition satisfaction depends on both perspectives. The goal is for the presuppositions of an expression to be assessed against the net local commitments imposed by the expression's local context.

(14) **Commitments**: the *commitments* of a context κ are $|\kappa| = \lceil \kappa \rceil \wedge \neg \lfloor \kappa \rfloor$.

Clearly, it makes sense to exclude worlds that do not satisfy the least upper bound, since they are guaranteed to be rejected, no matter how the utterance is continued.

[2] In addition, of course, quantification over the relevant semantic domains must be well-defined in the metalanguage.

It is less obvious that we should also exclude worlds that satisfy the greatest lower bound. The intuition, based on reasoning in Schlenker 2007 et seq., is that since these worlds are already guaranteed to make the larger computation true no matter how the utterance is continued, they can be safely ignored, since including them will not affect the final semantic value. In terms of presupposition satisfaction, it is safe to suppose that the negation of the lower bound holds. We'll see that this will be the key to making accurate predictions for examples involving disjunction and conditionals.

Since the reasoning underlying the definition of semantic commitments closely parallels Schlenker's construction of local contexts, it is natural to wonder to what degree the two notions are formally equivalent. The main difference is that Schlenker's definitions quantify over syntactic completions, and the definitions here quantify over semantic objects (see section 5 for discussion of why this difference is important). However, a large number of smaller differences stem from this main difference—so many that a thorough and rigorous proof characterizing the circumstances under which the two definitions are equivalent would be ponderous and full of unhelpful detail.

Nevertheless, a proof of equivalence for the propositional case can be informally sketched here. Since we're comparing a syntactic theory with a semantic theory, the argument will be easier to follow if we systematically deploy metonymy to conflate syntactic expressions with their denotations. For full details of Schlenker's definition see, e.g., Schlenker 2010a:385; for purposes of this sketch, the Schlenkerian local context of a proposition b occurring in a syntactic environment $a\_\_c$ is the smallest proposition p such that $a(p \text{ and } b')c' \leftrightarrow ab'c'$ for all well-formed completions b' and c'. That is, p contains all and only the worlds that can still potentially make a difference to the final result while holding a fixed. We need to show that p coincides with the semantic commitments of b's local context. If material in a rules out a world w no matter how a is completed, then w will not be in p, since by assumption w won't be in either $a(p \text{ and } b')c'$ or $ab'c'$. In other words, p is a subset of the least upper bound of b's local context. Likewise, if a rules in w no matter how the expression is completed, w will be in both $a(p \text{ and } b')c'$ and $ab'c'$, and so cannot distinguish between them. Thus p excludes the greatest lower bound of b's local context. For all remaining worlds w, there must be at least one completion that contains w (or else w would not be in the least upper bound), which means that w must be in p, from which it follows that p contains exactly the same worlds as the semantic commitments of b's local context.

We'll see in the next subsection that local contexts as defined here do indeed give results identical to Schlenker's definition in the core cases of the logical connectives. I'll also compare the present analysis to the predictions of Schlenker's approach at various points in what follows: for attitude verbs, see section 7.2; for quantifiers, see section 7.4; and for a discussion of certain problems involving linear order, see section 7.5.

4.2 Presupposition satisfaction

We can now consider the following theory of presupposition satisfaction:

(15) **Presupposition satisfaction**: the presupposition p of an expression with local context κ is satisfied just in case $|\kappa| \to p$, that is, just in case the commitments of κ entail p.

This is just Karttunen's strategy, given above in (1), expressed in terms of our continuized approach.

The following chart shows how the predictions of this hypothesis work out for some core cases involving unembedded logical connectives. Assume temporarily that these expressions are uttered in a maximally uninformative initial context, that is, when the initial continuation is the identity function.

Logical Form context	Local semantic context at '[ ]'	$\lceil\kappa\rceil$	$\lfloor\kappa\rfloor$	$\lvert\kappa\rvert$
A and [ ]	$\lambda p.A \wedge p$	A	$\bot$	A
A or [ ]	$\lambda p.A \vee p$	$\top$	A	$\neg A$
If A then [ ]	$\lambda p.A \to p$	$\top$	$\neg A$	A
Not [ ]	$\lambda p.\neg p$	$\top$	$\bot$	$\top$

Line by line: as explained above, the least upper bound of the local context at the right conjunct of a conjunction is the content of the left conjunct, A. This is because no matter what value p takes on, $A \wedge p$ entails A. Dually, if we choose $p = \bot$, $A \wedge p = A \wedge \bot = \bot$, so the least informative (largest) proposition that entails $A \wedge p$ for every choice of p is $\bot$. For disjunction, if we choose $p = \top$, $A \vee p = A \vee \top = \top$, so the most informative (smallest) proposition entailed by every choice of p is $\top$. Dually, as explained above, A entails $A \vee p$ for every choice of p, so the greatest lower bound for the local context at the right disjunct is A. For the conditional (approximated here, as in many discussions of dynamic semantics, as the material conditional), if we choose $p = \top$, $A \to p = A \to \top = \top$, so the least upper bound at the consequent is $\top$. No matter what we choose for p,

$\neg A$ entails $A \to p$, so the greatest lower bound is $\neg A$. For negation, if we choose $p = \bot$, $\neg p = \neg\bot = \top$, so the least upper bound is $\top$; and if we choose $p = \top$, the only thing that entails $\neg p = \neg\top = \bot$ is $\bot$ itself, so the greatest lower bound is $\bot$.

Given these values for $\lceil\kappa\rceil$ and $\lfloor\kappa\rfloor$, (14) and (15) taken together correctly predict that the presuppositions of the following sentences place no restriction on the initial context, except for the final example:

(16) It rained, and Ann knows that it rained.

(17) It rained, or Ann knows that it didn't rain.

(18) If it rained, Ann knows that it rained.

(19) Ann doesn't know that it rained.

In (16), the left conjunct rules out non-rain worlds, so when evaluating the right conjunct, it is safe to suppose that it is raining. In (17), the left disjunct rules in rain worlds, so when evaluating the right disjunct, it is safe to suppose that it didn't rain. In (18), the semantics of the conditional, along with the content of the antecedent, rules in non-rain worlds, so when evaluating the consequent, it is safe to suppose that it is raining. As for (19), since the semantic bounds imposed by negation are completely unconstraining, the only way the presupposition of the prejacent can be satisfied is if it is guaranteed by the initial context.

What happens when the initial context is not empty? Most theories model an initial context as a set of worlds, that is, an object of type `st`. Crucially, on the account here, the initial context, just like any context, is always a continuation, in this case a continuation with type `st` $\to$ `st`: a function from an (unlifted) utterance meaning to a final proposition. That means the initial context is not a plain proposition, and so falls under the recursive clause of the definitions of least upper and greatest lower bounds. What the bounds of the initial context turn out to be depends on what kind of function it is. If the initial context is the maximally uninformed context, $\lambda\kappa.\kappa$, then the only thing it is safe to suppose is $\top \wedge \neg\bot = \top$, which is appropriately uninformative. If the initial context is not empty, and the content of the utterance is added to the context set via intersection in the Stalnakerian conception of default update, then the initial context is a function that is equivalent to $\lambda p.A \wedge p$ for some choice of A, in which case it is a context in which it is safe to suppose $A \wedge \neg\bot = A$, which is also appropriate. In other words, in the default case, the update effect of a declarative is to conjoin the content of the utterance with the information in the initial context.

A simple example will help. Imagine that the initial context contains the information that today is Tuesday, and nothing else. Then the initial context is $\kappa_0 = \lambda p.\textbf{tuesday} \wedge p$, and the local commitments in this context are $|\kappa_0| = \lceil \kappa_0 \rceil \wedge \neg \lfloor \kappa_0 \rfloor = \textbf{tuesday} \wedge \neg \bot = \textbf{tuesday}$. That makes sense. The net result of uttering a token of *It's raining* in this context is $\boxed{\textbf{rain}}\ \kappa_0 = (\lambda \kappa.\kappa\, \textbf{rain})\ \kappa_0 = \kappa_0\ \textbf{rain} = \textbf{tuesday} \wedge \textbf{rain}$. So far, so good. If the sentence presupposes something that follows from it being Tuesday (*Fortunately, it's not Wednesday*), that presupposition will be satisfied by the context. But if the sentence presupposes instead something that is not entailed by it being Tuesday (e.g., *Fortunately, it's raining*), the presupposition is not guaranteed by the initial context, and the prediction is that uttering such a sentence in this context will create an instance of presupposition failure (albeit in this case one that would be particularly easy to repair through accommodation).

Note that the proposed treatment of disjunction accounts for the classic Partee bathroom sentence:

(20) Either this building doesn't have a bathroom, or it's in a funny place.

The greatest lower bound of the local context at the right disjunct is the proposition that this building doesn't have a bathroom. The negation of this bound is the proposition that this building *does* have a bathroom, which is precisely what is required to satisfy the existence presupposition of the pronoun *it*.

To sum up, the predictions of the semantic commitments of local contexts as defined here makes good predictions for the behavior of presuppositions in the presence of logical connectives. The predictions canvased in this section are the same as the predictions of Schlenker's Transparency theory, and (therefore) the same as Heim's 1983 predictions (after replacing her original CCP for disjunction with Beaver's 2001 refinement).

4.3 Lexical specification of presuppositions

Up to this point, lexical items have always entered the computation as lifted expressions that do not trigger presuppositions. Of course there must be lexical values that exploit the extra structure provided by the lifted computation. For instance, here is a presupposition-triggering lexical entry for *know*:

(21) know = $\lambda \kappa.\kappa(\lambda p : \underline{|\kappa| \rightarrow p}.\textbf{know}\ p)$

This expression uses the notation from Heim and Kratzer 1998 for specifying presuppositions: the function beginning with λp is defined only if the underlined condition between the ':' and the '.' is true, in this case, that $|\kappa| \rightarrow p$. So the verb phrase *know p* is defined only if the commitments of the local context guarantee p. If the presupposition is satisfied, then the at-issue value of the verb phrase is the relation **know** applied to p.

Crucially, in this lexical entry there are two bound occurrences of κ: once in the formula expressing the presupposition, and once in the larger enclosing formula expressing the at-issue truth conditions.

Turning now to epistemic modality, Mandelkern implements bounded modality as a presupposition. The connection is that we can think of presuppositions as a grammaticized condition on rational use. In the case of *know*, if the local context rules out rain, it would be irrational to contemplate whether Ann knows that it is raining. Likewise, in the modal case, in the same circumstances it would be irrational to suppose that *It might be raining* could be true.

In terms of the theory given here, Mandelkern's analysis looks like this:

(22) $\text{might} = \lambda\kappa.\kappa(\lambda pw : \underline{|\kappa|w \to \text{DOX}_w \subseteq |\kappa|}.\exists w' \in \text{DOX}_w.pw')$

On this theory, the at-issue truth conditions of *might p* are standard: *might p* is true at a world w just in case the epistemic accessibility relation DOX relates w to at least one accessible world at which p is true. The presupposition is that if the evaluation world w satisfies the commitments of the local context, all of its epistemically accessible alternatives must also do so. This is the sense in which the local context *bounds* epistemic modality. (This is the "weak" analysis of Mandelkern 2019 section 7.1, which is the analysis adopted in Mandelkern in prep. His strong analysis can just as easily be implemented instead, if desired.)

Given (22), (4) is a contradiction whenever defined. The reason is that in order for a world w to make the sentence as a whole true, it must be a non-rain world. But then w will satisfy the commitments of the local context of the right conjunct, so the lexical presupposition of *might* will require that all of the epistemic worlds accessible from w must also be non-rain worlds, which is exactly what it takes to make the *might* claim false.

Mandelkern argues that the local context for an epistemic modal should be symmetric rather than just the left context; see section 7.3 for discussion of how to implement symmetric local contexts if desired.

5 Pragmatics or semantics?

Is presupposition satisfaction a matter of pragmatics or semantics?

Schlenker 2007 argues in favor of pragmatics. He suggests that local presupposition satisfaction can be explained by the interaction of two conversational maxims. The first one, Be Articulate!, requires presuppositions to be explicitly expressed in the form of a conjunction. The second one, Be Brief!, forbids expressing content that is entailed by its local context. In case of conflict, Be Brief! wins. In combination, these maxims guarantee that one way or another, all presuppositions will be entailed by their local context. Schlenker calls this approach Transparency.

Transparency is a pragmatic theory. However, it is an unusual pragmatic theory. As Krahmer 2008:254 notes, Be Articulate! lacks motivation apart from presupposition satisfaction. In addition, in the spirit of other remarks of Krahmer, the result of violating a conversational maxim is usually suboptimal cooperativity, not infelicity, as it is in Transparency.

Beaver 2008:217 points out that the predictions of Transparency can be reconstructed without reference to maxims. The theory in Schlenker 2009 is just such a reconstruction: it simply places a necessary condition on the use of an expression that its presuppositions must be entailed by its local context. If that condition is not met, and the presupposition cannot be accommodated, the use is infelicitous.

Yet that theory retains an element of Transparency, in that it locates part of computing presupposition satisfaction outside of the semantic component. As in Transparency, determining whether a presupposition is entailed by a local context c depends on quantifying over all grammatical syntactic sentence completions, and then testing, for each completion, whether two closely related sentences are logically equivalent relative to c. Schlenker is consistent (e.g., 2007:328, 2008a:174, 2009:13, 2010a:385, 2010b:121) that the local context of an expression is supposed to depend only on features of its *syntactic* context.

However, Schlenker's definitions do not need to be stated in terms of syntax, and it is not clear that they ought to be. As Schlenker himself points out (2009:15), the relevant syntactic objects must contain arbitrarily complex recursive structure, so they are parts of trees, not strings. And despite the fact that only *grammatical* completions are allowed, grammaticality does no work in the system. That is, there does not appear to be any instance of a presupposition whose satisfaction hinges on a syntactic constraint on possible completions. So the syntactic aspect of the method is empirically idle. Furthermore, the atomic elements in the structures must be lemmas, not words, since assessing logical equivalence presupposes full disambiguation. For the same reason, quantifier scope must be fully resolved, as well as the distribution of coindexed pronouns, and so on. In other words, the definitions of local context must in effect be quantifying over *logical* *form* completions, not syntactic objects.

Suppose we abandon the appeal to syntax, and restate the theory entirely in terms of logical forms. We would have a theory expressed in terms of logical forms, with quantification over logical form completions, with logical equivalence between logical forms. In other words, this would be a semantic theory, just like the present proposal. This seems right: whether the presuppositions of an expression are satisfied depends entirely on the semantic content of the recursive compositional structure of the surrounding logical form. That is, it is purely a matter of entailment, and therefore purely a matter of semantics.

6 The conceptual necessity of continuations

A continuation is just a function from the basic (i.e., non-continuized) value of an expression to the result of a larger computation that depends on that value. This is why continuations are so well-suited to represent local semantic context.

Since every subexpression is part of a larger whole, it follows that every subexpression has a continuation with respect to that whole. In other words, continuations exist as a matter of conceptual necessity.

Arriving at a compositional theory of local contexts requires only that we figure out a method for composing continuations. By adapting continuation-passing style transforms from the theory of programming languages, that is precisely what this paper offers. To the extent that this provides an explanatory theory of local contexts, it supports the Continuation Hypothesis of Barker 2002:213, that "some linguistic expressions... have denotations that manipulate their own continuations."

Heim 1983:299 famously argues for "the conceptual priority of context change." For her, Context Change Potentials (CCPs) are "instructions specifying certain operations of context change. The CCP of *It is raining*, for instance, is the instruction to conjoin the current context with the proposition that it is raining." The reason that CCPs are supposed to be *prior* to truth conditions is that CCPs encode information about order of evaluation that is absent from the bare truth conditions. This means that truth conditions can be derived from CCPs, but not the other way around.

Some of the limitations of CCPs stem from the fact that they are functions on a set of worlds. Sets of worlds are a reasonable representation for the semantic content of a clause, but they do not generalize to other expression types. For instance, what instructions for updating the context could serve as the CCP of the attributive adjective *red*? The best we can do is calculate what *red* would have to denote in order to combine with other nearby expressions in order to construct a suitable CCP at the clausal level.

In order to arrive at a more uniformly incremental solution, on the continuized system here, expressions do not denote functions on sets of worlds. Instead, they denote functions on continuations. That is, a Heimean CCP has type $\mathtt{st} \to \mathtt{st}$, but a continuized clause here has type $(\mathtt{st} \to \mathtt{st}) \to \mathtt{st}$. For instance, the clause *it rained* is the function $\lambda\kappa.\kappa\,\mathbf{rain}$. Likewise, the continuized denotation of *red* is a function from an ordinary adjective continuation to the final result, namely, $\lambda\kappa.\kappa\,\mathbf{red}$, with type $(\mathtt{et} \to \mathtt{st}) \to \mathtt{st}$. These continuized types should remind you of generalized quantifiers; as mentioned above, generalized quantifiers have the type of a continuized individual.

This double twist—expressions denote functions from [functions on values to results] to the final result—is characteristic of continuation-passing style, starting with Plotkin 1975. This technique provides two distinct paths to a final result. On one path, the default path, the expression denotation simply applies its continuation to a suitable basic value, as just illustrated for *red* and *it rained*. The availability of this default strategy enables a systematic treatment of expressions that do not have any idiosyncratic dynamic properties—crucially, including the logical connectives, as discussed above. As for the second path, because lexical denotations take their own continuation as an argument, each lexical item ultimately has control over the final outcome. This allows lexical items to opt out of the default lifting operation, and to impose dynamic restrictions, such as presuppositions, or bounds on modal accessibility, or some other kind of dynamic effect.

Adding the double twist complicates the composition somewhat compared to Heim's system, but it solves the problem of dynamicizing the logical operators in a systematic way, without needing to craft special CCPs on a per-operator basis.

7 Issues and extensions

Presupposition projection has been intensively and extensively studied, and it would be impossible to address every problem that has been discussed in the literature. Nevertheless, in this section I will consider several major issues. In each case I'll suggest that the present proposal makes respectable default predictions, and that the ability of lexical values to override defaults provides welcome descriptive flexibility.

7.1 Epistemic modality as a test

Simon Goldstein (personal communication) points out that it is easy to reconstruct here a version of Veltman's well-known account of epistemic *might*. In Veltman 1996, *might* is an all-or-nothing update function, a 'test'. That is, depending on whether the prejacent is consistent with the input context, *might* either returns the entire input context unchanged, or else the empty context.

We can expess this analysis in the approach developed here as follows:

(23) $\text{might}_{\text{Veltman}} = \lambda\kappa.\kappa(\lambda pw.|\kappa| \wedge p \neq \bot)$

In (23), *might* takes its local context κ as an argument, and applies that context to a function that takes a prejacent p and returns a proposition, modeling propositions as a function from worlds to truth values. If the semantic commitments of the local context are consistent with the prejacent (i.e., if $|\kappa| \wedge p \neq \bot$), then the returned proposition maps all worlds to true; but if the local context is not consistent with the prejacent, the returned proposition maps every world to false.

I'll mention two points of interest. First, because checking for consistency in a Heim-style context update semantics requires checking the entire context set, a Veltman-style analysis requires the dynamic system to take entire contexts as inputs. As a result, it is a prime example of what Groenendijk and Stokhof 1990 call a non-distributive dynamic semantics, which means it cannot be re-engineered into an equivalent pointwise evaluation function that maps individual worlds to truth values. Perhaps surprisingly, the reconstruction here remains distributive: the net value of a sentence denotation in a context has type $\mathtt{s} \to \mathtt{t}$, a function from individual worlds to truth values. This is possible because, in addition to the individual evaluation world, we also have access to each expression's local semantic context. In this setting, $|\kappa|$ serves in the role of the local context set.

Second, as Goldstein points out, this value for *might*, just like in the Veltman analysis, does not trigger any presuppositions. Rather, it tests the local context for consistency with the prejacent, and returns either a tautology or a contradiction. Up to this point in the paper, local contexts have been used only in order to calculate presupposition satisfaction. Where defined, the semantic value of the logical forms is exactly what it would have been without any continuization. In contrast, the Veltmanesque analysis in (23) uses local contexts to determine the at-issue truth conditions.

We could modify (23) to make the test a presupposition. Then a sentence like *It's raining and it might not be raining* would have the status of a presupposition failure rather than a contradiction. Unfortunately, exploring this issue further will have to wait for another occasion.

7.2 Attitude verbs

Attitude verbs take full control over the local context of their complement.

(24) Ann wants Bill to stop smoking.

(25) Ann believes Bill smokes and wants him to stop.

As Heim 1992 points out, (24) does not presuppose that Bill smokes. Rather, it presupposes that Ann *believes* that he smokes. If the local context guarantees that Ann has this belief, as in (25), the sentence as a whole does not presuppose anything about Bill's habits. In the composition of (25), then, the initial conjunct satisfies the presupposition of *stop* indirectly: first, the conjunct constrains the beliefs of an agent, and then the presupposition-triggering expression is evaluated in the context of that agent's beliefs.

The analysis as developed so far will not give the desired result by default. If we treat *want* in a flat-footed way as undergoing the lexical lift rule, the local context of the embedded proposition *Bill to stop smoking* in (24) would be $\lambda p.\kappa[\textbf{ann}[\textbf{wants}\ p]]$. (Incidentally, Schlenker's Transparency theory makes a similar prediction, though the treatment there has a more subtle treatment of intensionality.) The commitments of this local context satisfy any presupposition that is entailed by each of Ann's desires. This is harmless, but it is not good enough. After all, Ann can believe something without wanting it. In fact, assuming that Ann believes that Bill smokes but does not want Bill to smoke, we'll incorrectly predict that the presuppositions of (24) will only be satisfied if the initial context entails that Bill smokes.

We must find a way to evaluate the presuppositions of the desire clause with respect to the *beliefs* of the subject of the attitude verb. This will be accomplished here by adjusting the denotation of the attitude verb. Similar to the way that a non-intersective adjective takes as its argument the intension of its nominal complement rather than its extension, we can make *want* a higher-order function, one that directly takes its continuized complement as an argument. Let $\hat{p}$ be a variable of type $(\texttt{st} \to \texttt{st}) \to \texttt{st}$, the sort of function that would be suitable as the denotation of the continuized proposition $\boxed{[\textbf{bill}\ [\textbf{stop smoking}]]}$. Then we can consider the following value for *wants*:

(26) wants $= \lambda \hat{p}\kappa.\kappa(\lambda x.\textbf{want}\ (\hat{p}(\lambda pw.pw \wedge w \in \text{DOX}_x))\ x)$

This *want* composes with its continuized complement via simple function application. It returns a (continuized) ordinary verb phrase meaning, which takes a verb phrase continuation κ and applies it to a function from an individual x to a truth value. That function will map an individual x to true just in case the attitude holder x stands in the **want** relation to a certain proposition representing the alleged desire. This desire proposition is computed by taking the continuized embedded clause $\hat{p}$ and applying it to a custom-built local context $f = \lambda pw.pw \wedge w \in \textsc{dox}_x$. No matter how p is instantiated, fp will entail $\lambda w.w \in \textsc{dox}_x$, which means that $|f| = \lambda w.w \in \textsc{dox}_x$. In other words, in the evaluation of (24), the local context of the desire clause will correspond exactly to the set of Ann's belief worlds. So (24) will presuppose only that Ann believes Bill smokes, and (25) will presuppose nothing about Bill's smoking habits, as desired.

7.3 Symmetric local contexts

Schlenker 2009 and Chemla and Schlenker 2012 offer a number of situations in which local presupposition satisfaction appears to work right to left.

(27) If Bill leaves too, Ann will agree to leave.

There is an interpretation of (27) on which the presupposition of *too* is satisfied by restricting attention to worlds in which the content of the consequent holds. One possible explanation is that under certain conditions, presuppositions can be satisfied by information anywhere in the sentence.

The idea that presupposition satisfaction might sometimes be symmetric is controversial. See Rothschild 2015 and Mandelkern et al. 2020 for discussion. Mandelkern et al. in particular provide experimental evidence that even if presupposition satisfaction might sometimes be symmetric, left to right evaluation is always available at least in the case of conjunction.

At a technical level, it is easy to switch from left to right evaluation to either right to left or fully symmetric, simply by replacing the rule for continuized function application given above in (7), repeated here as (28), with the symmetric rule in (29):

(28) $\boxed{\beta\ \gamma} = \lambda\kappa.\boxed{\gamma}(\boxed{\beta}(\lambda xy.\kappa[x\,y]))$ left-to-right continuized FA

(29) $\boxed{\beta\ \gamma} = \lambda\kappa.\boxed{\beta}(\lambda x(\boxed{\gamma}(\lambda y.\kappa[x\,y])))$ symmetric continuized FA

This second rule is identical to the continuization rule given in Barker 2002. (It was used there in search of a more explanatory theory of displaced scope; see

Barker and Shan 2014 for a more fully developed theory of scope.) Note that (29) is not merely a reversal of (28) that works right to left (such a strategy would also be possible, but implementing it is left as an exercise for the intrigued reader); rather, it gathers the commitments of all surrounding material on both the left and the right simultaneously.

Mandelkern 2019 argues for an analysis on which the bounds on epistemic modality are symmetric: not only must the local context of the right conjunct respect the commitments of the left conjunct, the local context of the left conjunct must respect the commitments of the right conjunct. He implements this by providing rules for computing local contexts on a per-lexical item basis, which is vulnerable to the criticisms of missed generalizations that apply to Heim's 1983 original update grammar. I am not convinced that modal bounds are fully symmetric in the way that Mandelkern claims. However, if a symmetric theory turns out to be desirable, the alternative continuization schema shows how to build a symmetric theory of bounded modality that does not have this vulnerability. That is, the system delivers symmetric local contexts without needing to make stipulations about individual lexical items.

It would be surprising if local contexts for ordinary presupposition satisfaction were typically left to right at the same time that bounded modality is always symmetric. If that turns out to be what the facts require, however, it is certainly feasible to have two layers of continuations, one which composes left to right, and one which composes symmetrically. For the purposes of this paper, however, I'll assume that local contexts are composed left to right by default, with symmetric composition as a processing alternative in certain situations.

7.4 Presuppositions in the scope of quantifiers

There is a debate about how presuppositions interact with quantification.

(30) Every student stopped smoking.

(31) Fewer than three students stopped smoking.

Some theories predict a uniformly existential presupposition (that at least one student used to smoke), some theories predict a uniformly universal presupposition (that all relevant students used to smoke), and yet other theories predict that the presupposition varies with the choice of the quantificational determiner. Chemla 2009 presents experimental evidence that supports the claim that the net presuppositions vary with the choice of quantificational determiner: (30) presupposes that every student used to smoke, but (31) presupposes something weaker, in the vicinity of a requirement that at least 3 students used to smoke.

It will not be possible to explore this issue in detail, but it is important to say what the default prediction of the present proposal is. In short, the default prediction is uniformly universal.

Here is how this prediction arises in more detail. The local context of *stopped smoking* in a sentence of the form *Q stopped smoking*, where *Q* is some quantificational phrase, is $\lambda P.\kappa(\mathbf{Q}\,P)$. Since everyone in every world has the property of being self-identical, and since no one in any world has the property of being non self-identical, the upper and lower bounds for this local context do not add or subtract any worlds from the worlds compatible with the initial context κ.

Now assume we have a lexical entry for *stopped* as follows:

(32) stop = $\lambda\kappa.\kappa(\lambda Px : \underline{|\kappa| \rightarrow \textsc{past}(P\,x)}.\mathbf{stop}\,P\,x)$

This value takes a verb phrase *P* and an individual *x* as arguments, and presupposes that the commitments of the local context κ entails that *P* applies to *x* in the past. So putting the verb phrase **stopped smoking** in the scope of a quantifier will trigger a presupposition about what happened in the past for each individual that the quantifier quantifies over.

By predicting universal projection, we're in good company: Heim 1983 and Schlenker 2007 et seq. also predict universal projection. However, the theory here has options that other theories do not. The universalist prediction is only the default prediction, assuming that quantifiers do not manipulate the local contexts of their semantic arguments. But of course, manipulating the environment within which their arguments are evaluated is exactly what quantifiers do: for instance, in *Everyone*$_i$ *likes his*$_i$ *mother*, the quantifier *everyone* evaluates its nuclear scope over and over again in contexts in which the variable x_i has been assigned to a different individual. So quantifiers in effect quantify over contexts (where each context is indexed by an individual). Nothing prevents us from constructing a value for a quantifier that explicitly manipulates the contexts with respect to which its arguments are evaluated.

Full exploration of this idea requires an ability to deal with relative clauses and bound pronouns, which in turn requires a treatment of Predicate Abstraction, which is given in the Appendix. Unfortunately, a suitably nuanced account of the interaction of quantification with presupposition will have wait for another occasion.

7.5 Timing issues and linear order

Anvari and Blumberg 2021 discuss examples in which a local context appears to anticipate material that follows the presupposition trigger.

(33) Both $\overset{a}{|}$ of John's eyes $\overset{b}{|}$ are open.

Deciding whether the presupposition of *both* is satisfied in (33) requires knowing how many eyes John has. The problem is that if we take the left-to-right bias of our theory of seriously, we must decide whether the presupposition of *both* has been satisfied based only on information already available at the point marked 'a', but we don't have the information we need to make that decision until we reach the point marked 'b'. Anvari and Blumberg suggest that the local context of the determiner therefore needs to contain the information carried by its restriction.

Schlenker 2008c:690 calls this issue "Barker's problem" (his example is *John awoke* $\overset{a}{|}$ *at 10am* $\overset{b}{|}$). He tentatively proposes that local contexts should be computed "only at points at which a presupposition trigger has been fed all its arguments (including adverbial ones)."

The problem dissolves on the account here. The resolution is that we do indeed determine the local context using only information prior to point 'a', but we compute the content of the presupposition using information up to point 'b'. Let's see how this works in detail. First, we'll compute the local context of *both* as per usual:

(34) Both eyes opened.

$$
\begin{aligned}
& \boxed{[[\textbf{both}\ [\textbf{eyes}]]\ \textbf{opened}]} \\
=& \lambda\kappa.\boxed{[\textbf{both eyes}]}\,(\lambda xy.\kappa[x\ y])\ \textbf{opened} && (7, 6, \beta) \\
=& \lambda\kappa.\big(\lambda\kappa'.\boxed{\textbf{both}}(\lambda x'y'.\kappa'[x'\ y'])\ \textbf{eyes}\big)\ (\lambda xy.\kappa[x\ y])\ \textbf{opened} && (7, 6, \beta) \\
=& \lambda\kappa.\boxed{\textbf{both}}(\lambda x'y'.\big(\lambda xy.\kappa[x\ y]\big)[x'\ y'])\ \textbf{eyes opened} && (\beta) \\
=& \lambda\kappa.\boxed{\textbf{both}}(\lambda x'y'y.\kappa[[x'\ y']\ y])\ \textbf{eyes opened} && (\beta)
\end{aligned}
$$

Applying the continuization strategy without any modfication, we see that $\boxed{\textbf{both}}$ (which is the continuation-level, presupposition-carrying denotation of *both*) takes three arguments: its local context, $\lambda x'y'y.\kappa[[x'\ y']\ y]$; its restritor, **eyes**; and its nuclear scope, **opened**. The local context contains only the initial context κ along with a functional skeleton of the compositional structure of the enclosing clause. As expected, the local context of *both* contains no lexical information to the right of location 'a'.

The next step is to find a suitable lexical denotation for *both*.

(35) both = $\lambda\kappa.\kappa(\lambda PQ : \underline{|\kappa| \to \mathbf{card}\ P = 2}.P \subseteq Q)$

The presuppositional part says that the semantic commitments of the local context must entail that the restriction P has cardinality 2. Once we substitute (35) into (34), after beta reduction we get the following value for the sentence as a whole:

(36) $\lambda\kappa.\kappa(: \underline{|\lambda x'y'y.\kappa[[x'\ y']\ y]| \to \mathbf{card\ eyes} = 2}.\mathbf{eyes} \subseteq \mathbf{opened})$

It is easy to show that the semantic commitments of $\lambda x'y'y.\kappa[[x'\ y']\ y]$ are equivalent to the semantic commitments of κ, so (36) says that (34) will be defined just in case the initial context entails that the number of relevant eyes is exactly 2, and the sentence will be true just in case the set of eyes is a subset of the set of opened things.

In other words, we get the desired result in a principled way without any stipulation. The local context respects strict left to right evaluation order (i.e., the order imposed by logical form), but the content of the presupposition depends on lexical values from later in the composition.

8 Conclusions

Heim 1983 takes clauses to denote continuations, that is, functions from contexts to the final result, type $\mathtt{st} \to \mathtt{st}$. This is a single-twist strategy, making clauses functions on their contexts rather than the other way around. This simple move gives a respectable theory of local presupposition satisfaction. However, it does not generalize to sub-clausal expressions, which requires stipulating update recipes for logical connectives, making the theory vulnerable to criticisms of insufficient explanatory power.

In Plotkin's 1975 Continuation Passing Style, expressions denote *functions on* continuations: functions from update functions to the final result, so that clauses have type $(\mathtt{st} \to \mathtt{st}) \to \mathtt{st}$. This double-twist strategy generalizes smoothly to subexpressions of any type, including logical connectives, allowing truly incremental dynamic update.

Inspired indirectly by de Groote's 2006 factoring of contexts into left context and right context, this paper offers an innovative continuation passing style grammar in which expressions are functions on their local context, that is, the initial context updated with the content of previously evaluated expressions.

Schlenker's 2007 Transparency reasoning—that what matters for presupposition satisfaction are only those situations that cannot be safely ignored—shows how to turn local contexts into upper and lower bounds on local semantic commitments, which in turn determines local presupposition satisfaction.

The resulting system is general, systematic, and fully incremental. In particular, logical expressions receive no special treatment, and have their ordinary bivalent denotations. In fact, the semantics of the underlying (unlifted) grammar are not disturbed in any way, as guaranteed by the Simulation theorem. All that the continuation passing reconfiguration does is make the implicit incremental semantic context of an expression explicitly available to it. The result is a truly minimal dynamic grammar on which a fully incremental dynamic computation depends on truth conditions, order of evaluation, and nothing else.

Appendix

This appendix extends the continuization method to logical forms that involve predicate abstraction. Predicate abstraction is a binding configuration created by Quantifier Raising. Following Heim and Kratzer 1998, I will write an instance of predicate abstraction as '$[i\ \beta]$', where i is an index (say, an integer) and β is a logical form containing somewhere within it a single instance of a co-indexed trace of the form 't_i'. So the index in effect binds the trace.

Because binding requires sensitivity to assignment functions, the pre-continuized types in this appendix will all be relativized to assignment functions in the usual way. So instead of a name having type $\mathtt{e}$, it will have type $\mathtt{a} \to \mathtt{e}$, where $\mathtt{a}$ is the type of an assignment function. Likewise, a one-place predicate of type $\mathtt{e} \to \mathtt{t}$ will have type $\mathtt{a} \to \mathtt{e} \to \mathtt{t}$, and so on. Then for β with type $\mathtt{a} \to \mathtt{A} \to \mathtt{B}$ and γ with type $\mathtt{a} \to \mathtt{A}$, the assignment-aware function application of β to γ is $\lambda g.\beta g\ (\gamma g)$. That is, the functor and its argument are each evaluated with respect to the assignment function of the composed expression. Finally, the assignment-aware abstraction $[i\ \beta]$ is $\lambda gx.\beta g^{i \mapsto x}$: a function from an assignment and an individual x to the result of evaluating β with respect to $g^{i \mapsto x}$, which is a modified version of the assignment function g that maps the index i to x.

Given these assumptions, here is the method for continuizing an arbitrary logical form:

$$\boxed{\alpha} = \lambda\kappa.\kappa(\lambda g.\alpha g) \qquad \text{(lexical item)}$$

$$\boxed{[\beta\ \gamma]} = \lambda\kappa.\boxed{\gamma}(\boxed{\beta}(\lambda xy.\kappa(\lambda g.[xg\ (yg)]))) \qquad \text{(function application)}$$

$$\boxed{[i\ \beta]} = \lambda\kappa.\boxed{\beta}(\lambda b.\kappa(\lambda gx.bg^{i\mapsto x})) \qquad \text{(predicate abstraction)}$$

$$\boxed{t_i} = \lambda\kappa.\kappa(\lambda g.g\ i) \qquad \text{(trace)}$$

Figure 1: Continuizing an arbitrary logical form. This scheme delivers local contexts that contain all and only the content of previously-evaluated expressions.

In order to see that this is indeed a faithful extension of the scheme given above, note that the lexical rule is eta-equivalent to (6), and the rule for function application differs from (7) only as required to implement assignment-aware function application.

This extended continuization scheme enjoys Flip and Simulation.

Lemma: Flip: for all logical forms α, $\boxed{\alpha} = \lambda\kappa.\kappa\alpha$. Base case: true for lexical items and traces. Inductive case: assume Flip holds for β and γ. Then $\boxed{[\beta\ \gamma]} = \lambda\kappa.\boxed{\gamma}(\boxed{\beta}(\lambda xy.\kappa(\lambda g.[xg\ (yg)]))) = \lambda\kappa.\kappa(\lambda g.[\beta g\ (\gamma g)])$, so Flip holds for function application. Finally, $\boxed{[i\ \beta]} = \lambda\kappa.\boxed{\beta}(\lambda b.\kappa(\lambda gx.bg^{i\mapsto x})) = \lambda\kappa.\kappa(\lambda gx.\beta g^{i\mapsto x})$, so Flip holds for predicate abstraction.

Corrolary: Simulation: for all logical forms α, $\boxed{\alpha}\,\mathrm{I} = \alpha$.

References

Anvari, Amir and Kyle Blumberg. 2021. Subclausal local contexts. *Journal of Semantics*. doi:10.1093/jos/ffab004

Barker, Chris. 2002. Continuations and the Nature of Quantification. *Natural Language Semantics* 10: 211–242. https://doi.org/10.1023/A:1022183511876

Barker, Chris and Chung-chieh Shan. 2014. *Continuations and Natural Language*. Oxford University Press.

Beaver, David. 2001. *Presupposition and Assertion in Dynamic Semantics*. CSLI Publications.

Beaver, David. 2008. As brief as possible (but no briefer). *Theoretical Linguistics* 34.3: 213-228.

Chemla, Emmanuel. 2009. Presuppositions of quantified sentences: experimental data. *Natural Language Semantics* 17(4). 299–340.

Chemla, Emmanuel and Philippe Schlenker. 2012. Incremental vs. symmetric accounts of presupposition projection: an experimental approach. *Natural Language Semantics* 20:177–226. https://doi.org/10.1007/s11050-012-9080-7

Chung, WooJin. In prep. Local context is calculated domain by domain, for each maximal projection.

de Groote, Philippe, 2006. Towards a Montagovian Account of Dynamics. In M. Gibson and J. Howell (eds). *Semantics and Linguistic Theory* 16:1–16. Available at: http://dx.doi.org/10.3765/salt.v16i0.2952.

de Groote, Philippe and Ekaterina Lebedeva. 2010. Presupposition accommodation as exception handling. In *Proceedings of the SIGDIAL 2010 Conference*.

George, B. R. 2014. Some remarks on certain trivalent accounts of presupposition projection. *Journal of Applied Non-Classical Logics* 24.1–2: 86–117. http://dx.doi.org/10.1080/11663081.2014.911521

Groenendijk, Jeroen and Martin Stokhof. 1990. Two theories of dynamic semantics. ITLI Prepublications. Published in 2005 in Jan van Eijck (ed). *Logics in AI. Lectures notes in computer science* (Vol. 478). 55–64. doi:10.1007/BFb0018433

Groenendijk, Jeroen, Martin Stokhof, and Frank Veltman. 1996. Coreference and modality. In Shallom Lappin (ed). Handbook of contemporary semantic theory. 179–213. Oxford: Blackwell. doi:10.1111/b.9780631207498.1997.00010.x

Heim, Irene. 1983. On the Projection Problem for Presuppositions. Presented in 1983 at WCCFL 2; published in the proceedings of WCCFL in 1988; reprinted in Paul Portner and Barbara Partee (eds). 2008. *Formal Semantics*: 249–260. https://doi.org/10.1002/9780470758335.ch10

Heim, Irene. 1992. Presupposition Projection and the Semantics of Attitude Verbs. *Journal of Semantics* 9(3). 183–221. doi:10.1093/jos/9.3.183.

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell.

Karttunen, Lauri. 1973. Presuppositions of compound sentences. *Linguistic Inquiry* 4.2: 169–193.

Karttunen, Lauri. 1974. Presupposition and linguistic context. *Theoretical Linguistics* 1: 181—194.

Krahmer, Emiel. 2008. Why Be Articulate? Two ways to look at the Transparency Theory *Theoretical Linguistics* 34.1: 253-259.

Lebedeva, Ekaterina. 2012. *Expressing Discourse Dynamics via Continuations*. PhD Dissertation, University of Lorraine, France.

Mandelkern, Matthew. 2019. Bounded Modality. *The Philosophical Review* 128.1: 1–61. https://doi.org/10.1215/00318108-7213001

Mandelkern, Matthew. In prep. [book]

Mandelkern, Matthew, Jérémy Zehr, Jacopo Romoli, and Florian Schwarz. 2020. We've discovered that projection across conjunction is asymmetric (and it is!). *Linguistics and Philosophy* 43:473–514. https://doi.org/10.1007/s10988-019-09276-5

Partee, Barbara Hall. 1987. Noun phrase interpretation and type-shifting principles. In Jeroen Groenendijk, Dick de Jongh, and Martin Stokhof (eds). *Studies in discourse representation theory and the theory of generalized quantifiers*: 115–143. Groningen-Amsterdam Studies in Semantics 8, Dordrecht: Foris.

Plotkin, Gordon D. 1975. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science* 1.2: 125–159.

Rothschild, Daniel. 2015. Explaining presupposition projection with dynamic semantics. *Semantics and Pragmatics* 4.3: 1–43. https://doi.org/10.3765/sp.4.3

Schlenker, Philippe. 2007. Anti-dynamics: presupposition projection without dynamic semantics. *Journal of Logic, Language, and Information* 16. 325–356. https://doi.org/10.1007/s10849-006-9034-x

Schlenker, Philippe. 2008a. Be articulate: A pragmatic theory of presupposition projection. *Theoretical Linguistics* 34.3, 157–212.

Schlenker, Philippe. 2008b. Presupposition projection: Explanatory strategies. *Theoretical Linguistics* 34, 287–316.

Schlenker, Philippe. 2008c. Presupposition Projection: the New Debate. In T. Friedman and S. Ito (eds). *Semantics and Linguistic Theory (SALT)* 18: 655–693. http://dx.doi.org/10.3765/salt.v0i0.2503.

Schlenker, Philippe. 2009. Local Contexts. *Semantics & Pragmatics* 2.3: 1–78. `https://doi.org/10.3765/sp.2.3`

Schlenker, Philippe. 2010a. Presuppositions and Local Contexts *Mind* 119.474: 377-391.

Schlenker, Philippe. 2010b. Local contexts and local meanings. *Philosophical Studies* 151.1: 115-142.

Shan, Chung-chieh and Chris Barker. 2006. Explaining Crossover and Superiority as Left-to-right Evaluation. *Linguistics and Philosophy* 29.1: 91-–134. http://dx.doi.org/10.1007/s10988-005-6580-7

Soames, Scott. 1989. Presupposition. In D. Gabbay and F. Guenthner (eds). *Handbook of philosophical logic* Volume IV, Chapter 9: 553–616. Dordrecht: Reidel.

Stalnaker, Robert. 1973. Presuppositions. *Journal of Philosophical Logic* 2.4: 447-457.

Veltman, Frank. 1996. Defaults in Update Semantics. *Journal of Philosophical Logic* 25.3: 221-261.

Wittgenstein, Ludwig. 1958[1953]. *Philosophical Investigations*. 2d edition. Translated by G. E. M. Anscombe. Basil Blackwell.