Composing local contexts

Chris Barker, *NYU Linguistics*

June 15, 2021

DRAFT, comments welcome

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 minimal dynamic semantics in which the only thing that is dynamically tracked is the semantic content of what has already been said.

Acknowledgments: Thanks for discussion to WooJin Chung, Matthew Mandelkern, and Philippe Schlenker. Historical note: I circulated a manuscript presenting the basic technique in 2008, which I made available on `semanticsarchive.net` in 2011. That manuscript is the work cited in Schlenker 2010b as Barker 2008.

1 Towards a minimal 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 pioneered in 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 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, 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 (though also see the discussion of symmetric local contexts below in section 6). 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. 1995 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 of epistemic modality 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 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.

In order to track incremental semantic composition, we're going to lift an ordinary logical form into a continuation-passing style computation.[1] What makes

[1] In addition to my own earlier work on continuations, including Barker 2002 and Shan and

continuation-passing style useful here, as we will see, is that it provides explicit semantic access to implicit semantic context. I'll say more about these conceptual underpinnings below in section 8, but it will be helpful to see the approach in action first.

We need to consider three LF configurations: lexical items, function application, and predicate abstraction. The discussions of the first two configurations are immediately below. There is nothing problematic about predicate abstraction, but we won't need it to illustrate how the system works, so those details appear in a brief appendix.

For a lexical item α that does not trigger any presuppositions, the lifted version of α, which I'll write as $\boxed{\alpha}$, is simple:

(6) $\boxed{\alpha} = \lambda\kappa.\kappa\alpha$ Lifted lexical item (presupposition-free)

Here and throughout, I'll use the variable κ to stand for the current local context. So the lifted version of α takes κ and applies it to the ordinary (unlifted) value of α.

Because the semantic type of lexical items varies, the type of κ will also vary; that is, the lifting operation is polymorphic. Note that this lexical lifting operation is just a generalized (i.e., polymorphic) version of Partee's 1987 LIFT type-shifter, which turns an individual-denoting expression into a generalized quantifier. For example, if **ann** is an individual constant of type $\mathtt{e}$, then $\boxed{\mathbf{ann}} = \text{LIFT}(\mathbf{ann}) = \lambda\kappa.\kappa\,\mathbf{ann}$, the generalized quantifier corresponding to the name *Ann*.

The lifted version of an ordinary instance of function application looks like this:

(7) $\boxed{\beta\ \gamma} = \lambda\kappa.\boxed{\gamma}(\boxed{\beta}(\lambda xy.\kappa[xy]))$ Lifted function application

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

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.

Let's immediately consider a simple example.

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

The end result is something very close to the value of the original (unlifted) logical form. The only difference is the presence of the explicit continuation κ in the final line. This κ corresponds to the semantic context within which the expression *Ann laughed* is being evaluated. In order to recover the standard truth conditions of the sentence, we can simply apply the lifted result to the identity function $\text{I} = \lambda\kappa.\kappa$, which represents the neutral context, like this: $(\lambda\kappa.\kappa[\textbf{ann laughed}])\ \text{I} = \text{I}\ [\textbf{ann laughed}] = [\textbf{ann laughed}]$.

This point is worth emphasizing: the continuized composition encodes the exact same denotation as the uncontinuized logical form, except for the final application to the identity function. In Plotkin's 1975 paper introducing continuation passing style transforms, he proves a theorem he calls Simulation: for any expression α, $\boxed{\alpha}\ \text{I} = \alpha$. Simulation holds for the continuization scheme here. To see why, consider a lemma I'll call Flip: for all logical forms 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. In other words, continuization is just a different way of arranging the same compositional elements, and does not change the semantic value of a logical form.

Simulation only holds, of course, until we add lexical items that trigger presuppositions. Those denotations, as we'll see, do not correspond to the lifted values of any non-continuized expression. Instead, they take advantage of the special compositional flexibility afforded by continuization. The situation is analogous to intensionalizing an extensional grammar: the lifted (intensionalized) grammar computes meanings that are identical to the original extensional grammar (identical up to application of the final intension to an evaluation world), until we add lexical items that exploit intensionality (see Ben-Avi and Winter 2007 for a discussion of intensionalization along these lines).

We're now ready to consider some simple examples involving presuppositions. Let's start with the first element of the minimal pair discussed above in (2).

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

b. $\boxed{[\textbf{rained}\ [\textbf{and}\ [\textbf{ann knows-it-rained}]]]}$

c. $= \lambda\kappa.\boxed{\textbf{knows-it-rained}}\ (\lambda P.\kappa(\textbf{and}\ (P\ \textbf{ann})\ \textbf{rained}))$

In this example, the verb phrase *knows it rained* presupposes that it rained. Assume for now that $\boxed{\textbf{knows-it-rained}}$ is the presupposition-aware denotation of the verb phrase, whatever that turns out to be, and that the type of this denotation is suitable for participating in a continuized computation. (We'll see shortly in section 5.1 how to compute that denotation compositionally.)

Recall that κ in the final line corresponds to the initial context. Then the argument to the presupposition-triggering verb phrase *knows it rained* is $\lambda P.\kappa(\textbf{and}\ (P\ \textbf{ann})\ \textbf{rained})$: a function from a verb phrase meaning, P, to the result of updating the initial context with the proposition that P applies to Ann, and it rained. Echoing Schlenker, no matter what value P takes, the updated context will entail that it rained. (I'll make this reasoning precise shortly.) Thus the argument accurately characterizes the local context of the second conjunct, since it contains all of the information necessary to conclude that the presupposition that it rained has been satisfied.

Here is the second member of the minimal pair discussed above in (3) for comparison:

(9) a. Ann knows it rained, and it rained.

b. $\boxed{\textbf{ann knows-it-rained}\ (\textbf{and rained})}$

c. $= \lambda\kappa.\boxed{\textbf{knows-it-rained}}\ (\lambda P f.\kappa(f(P\ \textbf{ann})))\ (\textbf{and rained})$

In this case, the argument to the presuppositional verb phrase *knows it rained* is $\lambda P f.\kappa(f(P\ \textbf{ann}))$: a function from a verb phrase meaning, P, and a one-place truth-value operator f to $\kappa(f(P\ \textbf{ann}))$. Once we finish evaluating the rest of the sentence, the function f will turn out to be **and rained**, but that information does not appear in the first argument of the lifted verb phrase. Thus the information contained in the first argument correctly predicts that the presupposition triggered by the verb can only be satisfied if it is entailed by the initial context κ.

The correspondence between the semantic argument of a lifted expression and its local context is systematic.

(10) **Local semantic context**. The first argument of every continuized expression is always its local semantic context.

In the system here, the first argument of a continuized expression contains the semantic content of all and only the LF material to the left of the expression in question. This 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 exactly the semantic content of its left context.

Some important points to note right off the bat:

- The lifted computation is purely semantic, and, unlike Schlenker 2007, 2009, etc., 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 lifting function application given in (7).

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

4 Turning contexts into entailments

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 (8), namely, $\lambda P.\kappa(\textbf{and}\ (P\ \textbf{ann})\ \textbf{rained})$ (recall that κ here represents the initial context for the relevant utterance of (8)). 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 $f = \lambda p.\mathbf{rain} \wedge p$, 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 $f = \lambda p.\mathbf{rain} \vee p$, rain worlds will entail fp no matter how p is instantiated. So this 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:

$$\lceil f \rceil = \begin{cases} f & \text{if } f \text{ has type } \texttt{st} \\ \exists x_{\texttt{a}}.\lceil fx \rceil & \text{if } f \text{ has type } \texttt{a} \to \texttt{b} \end{cases} \tag{11}$$

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

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.

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

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

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

5 Presupposition satisfaction

We can now consider the following theory of presupposition satisfaction:

(14) **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 local 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 simple 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.

LF context	$\kappa =$ local context at '[ ]'	$\lceil\kappa\rceil$	$\lfloor\kappa\rfloor$	$\|\kappa\|$
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 $\bot$.

Given these values for $\lceil\kappa\rceil$ and $\lfloor\kappa\rfloor$, (13) and (14) taken together correctly predict that the presuppositions of the following sentences place no restriction on the initial context, except for the final example:

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

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

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

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

In (15), the left conjunct rules out non-rain worlds, so when evaluating the right conjunct, it is safe to suppose that it is raining. In (16), the left disjunct rules in rain worlds, so when evaluating the right disjunct, it is safe to suppose that it didn't rain. In (17), 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 (18), 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 $\mathtt{st} \to \mathtt{st}$: 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 that it is raining (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 one that would be particularly easy to repair through accommodation).

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

(19) 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*.

5.1 Lexical specification of presuppositions

Up to this point, lexical items have always entered the computation as lifted expressions that do not trigger presuppositions. 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 trigger a presupposition. For instance, here is a presupposition-active lexical entry for *know*:

(20) Lexical entry for *know*: $\lambda\kappa.\kappa(\lambda p : |\kappa| \to p.\textbf{know}\ p)$

This expression uses the notation from Heim and Kratzer 1998 for specifying presuppositions: the function beginning with λp is defined only if the condition between the ':' and the '.' is true, in this case, $|\kappa| \to p$. So the verb phrase *know*

p is defined only if the commitments of the local context guarantee p. If the presupposition is satisfied, the 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 formula expressing the truth conditions of the expression as a whole.

Turning now to epistemic modality, Mandelkern implements bounded modality as a presupposition. We can think of presuppositions in general as a condition on rational use: if the local context at a certain point in the composition of an utterance rules out rain, it would be irrational to contemplate whether Ann knows that it is raining at that point. Likewise, in that same context, 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:

(21) might $= \lambda\kappa.\kappa(\lambda pw : |\kappa|w \rightarrow \textsc{dox}_w \subseteq |\kappa|.\exists w' \in \textsc{dox}_w.pw')$

On this theory, the 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 an 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. The strong analysis can just as easily be implemented instead, if desired.)

Given (21), (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 prefers an analysis on which (4) and (5) have the same explanation. He achieves this by requiring the relevant local contexts to be 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. The next section explains how to build a symmetric theory of bounded modality that does not have this vulnerability: it is fully systematic, just by adjusting the rule for lifted function application.

6 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.

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

There is an interpretation of (22) on which the presupposition of *too* is satisfied by the content of the consequent. 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. We simply need to replace the rule for dynamicizing function application given above in (7), repeated here as (23), with the symmetric rule in (24):

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

(24) $\boxed{\beta\ \gamma} = \lambda\kappa.\boxed{\beta}(\lambda x(\boxed{\gamma}(\lambda y.\kappa[xy])))$ 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 (24) is not merely a reversal of (23) that works right to left (such a strategy would also be possible, but is left for an exercise for the intrigued reader); rather, it gathers the commitments of all surrounding material on both the left and the right simultaneously.

As mentioned in the previous section, Mandelkern proposes that bounded modality always depends on symmetric local contexts. 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 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 that 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, the 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 (p. 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.

8 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 formal 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 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 of sets of worlds. Instead, they denote functions on continuations. So 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 $\lambda\kappa.\kappa\,\mathbf{rained}$. 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 dynamic properties—crucially, including the logical connectives, as discussed above. As for the second path, because denotations take their continuation as an argument, the denotation 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.

9 Conclusions

Heim 1983 takes clauses to denote continuations, more specifically, functions from contexts to the final result, type $\texttt{st} \to \texttt{st}$. 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 $(\texttt{st} \to \texttt{st}) \to \texttt{st}$. This double-twist strategy generalizes smoothly to subexpressions of any type, including logical connectives, allowing truly incremental dynamic update.

Inspired in part 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 semantic context of an expression explicitly available to it. The result is a truly minimal dynamic grammar on which a fully incremental dynamic update computation depends on truth conditions, order of evaluation, and nothing else.

Appendix

This appendix explains in more detail how to continuize a logical form. The method here is more complete than the one in the text in two notable ways: it handles predicate abstraction, as promised, and it explicitly manages the direction of function application depending on the types of the logical form elements.

It will be helpful to define some combinators. $\textsc{i}, \textsc{b}, \textsc{c}$ first appeared in Shönfinkel 1924, but the names here are as in Curry 1930:513; $\textsc{t}$ (the Thrush bird), as far as I know, was first named by Smullyan 1985.

$$\textsc{i} = \lambda x.x \qquad \textsc{t} = \lambda xy.yx \qquad \textsc{b} = \lambda xyz.x(yz) \qquad \textsc{c} = \lambda xyz.xzy \tag{25}$$

In addition to these standard combinators, we will need one special-purpose combinator, $\textsc{h}$:

$$\textsc{h} = \lambda flrk.r(l(\lambda xy.k(fxy))) \tag{26}$$

This is the same treatment of function application given in (7), except parameterized for the direction of function application (as explained below).

With these tools in hand, we can define a function (‘$\boxed{\cdot}$’) that recursively maps an ordinary logical form into a continuized logical form.

LLC

$$\boxed{c} = \mathrm{T}\ c \qquad (c \text{ is a lexical item})$$

$$\boxed{\beta_{\mathtt{ba}}\ \gamma_{\mathtt{b}}} = \mathrm{H\ I}\ \boxed{\beta}\ \boxed{\gamma}$$

$$\boxed{\beta_{\mathtt{b}}\ \gamma_{\mathtt{ba}}} = \mathrm{H\ T}\ \boxed{\beta}\ \boxed{\gamma}$$

$$\boxed{i\ t_i} = \mathrm{T\ I} \qquad (i \text{ is an index, } t_i \text{ a trace})$$

$$\boxed{i\ [\beta_{\mathtt{ba}}\ \gamma_{\mathtt{b}}]} = \mathrm{H\ B}\ \boxed{\beta}\ \boxed{i\ \gamma} \qquad (t_i \text{ occurs in } \gamma)$$

$$\boxed{i\ [\beta_{\mathtt{b}}\ \gamma_{\mathtt{ba}}]} = \mathrm{H}\ (\mathrm{C\ C})\ \boxed{\beta}\ \boxed{i\ \gamma} \qquad (t_i \text{ occurs in } \gamma)$$

$$\boxed{i\ [\beta_{\mathtt{ba}}\ \gamma_{\mathtt{b}}]} = \mathrm{H\ C}\ \boxed{i\ \beta}\ \boxed{\gamma} \qquad (t_i \text{ occurs in } \beta)$$

$$\boxed{i\ [\beta_{\mathtt{b}}\ \gamma_{\mathtt{ba}}]} = \mathrm{H}\ (\mathrm{C\ B})\ \boxed{i\ \beta}\ \boxed{\gamma} \qquad (t_i \text{ occurs in } \beta)$$

Figure 1: The Left Local Context transform. '$\beta_{\mathtt{ba}}$' must have type $\mathtt{b} \to \mathtt{a}$ for some types $\mathtt{b}$ and $\mathtt{a}$. For any logical form α, $\boxed{\alpha}$ is a continuized logical form satisfying Flip ($\boxed{\alpha} = \lambda\kappa.\kappa\alpha$) and Simulation ($\boxed{\alpha}\ \mathrm{I} = \alpha$).

There are two rules for function application, distinguished by the order of functor and argument as indicated by the typing subscripts. There are five rules for predicate abstraction. One is for handling the trace, t_i. The remaining four differ in whether the coindexed trace is within β or γ, as well as by the order of functor and argument.

It is easy to complete the proof of Simulation, by proving that Flip holds for predicate abstraction. Recall that Flip says that $\boxed{\alpha} = \lambda\kappa.\kappa\alpha$ for all expressions α. For the base case, Flip holds for the clause involving traces by inspecting the definition of T. For the first recursive clause for predicate abstraction, we have

$$\begin{aligned}
\boxed{i\,[\beta_{\mathtt{b}\to\mathtt{a}}\,\gamma_{\mathtt{a}}]} &= \textsc{h}\ \textsc{b}\ \boxed{\beta}\ \boxed{i\,\gamma} && (t_i \text{ occurs in } \gamma)\\
&= \lambda\kappa.\boxed{i\,\gamma}(\boxed{\beta}(\lambda xy.\kappa(\textsc{b}xy))) && (\text{definition of } \textsc{h})\\
&= \lambda\kappa.\kappa(\textsc{b}\beta[i\,\gamma]) && (\text{inductive hypothesis for } \boxed{\beta} \text{ and } \boxed{i\,\gamma})\\
&= \lambda\kappa.\kappa(\lambda x.\beta([i\,\gamma]\,x)) && (\text{definition of } \textsc{b})\\
&= \lambda\kappa.\kappa(i\,[\beta\,\gamma]) && (\text{since } t_i \text{ occurs in } \gamma)
\end{aligned}$$

Proving Flip for the other three clauses is analogous.

Example:

(27) It was raining, and everyone who knew it was raining left.

(28) Logical form: [rain [and [[everyone [1 [t_1 [knew rain]]]] left]]]

(29) TCs: $\lambda\kappa.\kappa$ (and (everyone (λx. knew rain x) left) rain)

```
H T (T rain)
    (H I (T and)
         (H I (H I (T everyone)
                   (H (C B) (T I)
                            (H I (T knew) (T rain))))
              (T left)))
```

(30) Everyone that Ann knew had left called.

(31) Logical form: [[everyone [1 [ann [knew [t_1 left]]]]] called]

(32) Continuized logical form:

```
H I (H I (T everyone)
         (H (C C) (T ann)
                 (H B knew
                      (H (C B) (T I)
                               (T left)))))
    (T called)
```

(33) TCs: $\lambda\kappa.\kappa$ (everyone (λx. knew (left x) ann) called)

(34) a. Local context for *knew*:
b. $(\lambda RPQ.\kappa(\textbf{everyone}\ (\lambda x.R(Px)\ \textbf{ann})Q))$
c. Presupposition, after incorporating the complement of *know*: for each person x, that x left.

Reasonable quantified presupposition. If too strong, one strategy would be to allow the implicit domain of quantification of *everyone* to be restricted to some set of relevant people, perhaps those who left, i.e., the good candidates for people that Ann might have known left. There may very well be other strategies for weakening quantificational presuppositions.

References

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

Ben-Avi, G., and Yoad Winter. 2007. The semantics of intensionalization. In R. Muskens (ed). *Workshop on new directions in type-theoretic grammars* 98–112.

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.

Curry, Haskell. 1930. Grundlagen der Kombinatorischen Logik. *American Journal of Mathematics* 52.3: 509-536.

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. 1991. Dynamic Predicate Logic. *Linguistics and Philosophy* 14: 39-100. https://doi.org/10.1007%2Fbf00628304

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

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

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 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. 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.
Schönfinkel, Moses. 1924. Über die Bausteine der mathematischen Logik. *Mathematische Annalen* 92.3–4: 305–316. doi:10.1007/bf01448013. Translated by Stefan Bauer-Mengelberg as ”On the building blocks of mathematical logic” in Jean van Heijenoort, 1967. *A Source Book in Mathematical Logic*, 1879–1931. Harvard University Press, pp. 355–66.
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
Smullyan, Raymond. 1985. *To Mock a Mockingbird*. Knopf.
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.