

Making Dynamics Semantics Explanatory: Presupposition Projection

Daniel Rothschild*

September 11, 2008

Abstract

Understanding the pattern by which complex sentences inherit the presuppositions of their parts (*presupposition projection*) has been a major topic in formal pragmatics since the 1970s. Heim’s classic paper “On the Projection Problem for Presuppositions” (1983) proposed a replacement of truth-conditional semantics with a dynamic semantics that treats meanings as instructions to update the common ground. Heim’s system predicts the basic pattern of presupposition projection quite accurately. The classic objection to this program (including other versions of dynamic semantics) is that the treatment of binary connectives is stipulative, and other, equally natural treatments fail to make the right predictions about presupposition projection. I give a variation on Heim’s system that is designed to escape this objection. I show that the most liberal possible version of this variant is equivalent to a strong-Kleene system in terms of its definedness conditions.

1 Introduction

Presupposition projection is the pattern by which complex sentences inherit the presuppositions of their parts. Some of the basic contours of this pattern have long been understood, but there has been a struggle to explain why presuppositions project in the particular way they do. A

*I am grateful to Be Birchall, Emmanuel Chemla, Haim Gaifman, Nathan Klinedinst, Benjamin Spector, Philippe Schlenker and workshop audiences at the ENS and the University of Chicago for their helpful comments.

good story should both given an account of what linguistic presuppositions are in combination with an account of the semantics and/or processing of complex sentences that yields as a result the pattern of presupposition projection. One of the most well-regarded efforts to do this is the dynamic approach to semantics developed by Heim (1982) and Kamp (1981).

Perhaps the most serious challenge to the dynamic framework as a means of explaining presupposition projection is that certain aspects of its semantics tend to involve stipulations about the meanings of the individual logical operators that are a) are implausible in their own right and b) make the theory insufficiently explanatory of the pattern of presupposition projection. The heart of this paper, starting in section 6 and laid out explicitly in section 11, is a reformulation of dynamic semantics that escapes this criticism. I start with a discussion of the basic empirical facts of presupposition projection.

2 Presuppositions

The linguistic phenomenon of presupposition is illustrated by these two pairs of sentences.

- (1) a. John stopped smoking
b. John used to smoke.
- (2) a. John knows that Ted lied
b. Ted lied.

Without concerning ourselves with the exact the relationship between the a. and b. sentences, it will suffice it to say that in both cases the a. sentence, when used, seems to imply that the speaker is taking the b. sentence for granted.¹ Following the literature, we'll call this sort of relationship one of presupposition without thereby making any serious assumption about what this ultimately amounts to. (We'll also call all the b. sentences presuppositions of the a. sentences.)

This phenomenon persists under negation in most cases. For instance, just as with a use of (1-a), a use of (3) normally takes (1-b) for granted.²

¹For background on presupposition, see Soames (1989), Heim (1990) or Beaver (2001).

²I say "normally" to put aside cases of cancellation. Presuppositions are clearly only defaults and are often overridden. Ultimately one must give a story of such cancellation, but I'm going to put that issue aside here.

(3) John didn't stop smoking.

The general term for the ability of presuppositions to persist through negation (and in other embedded contexts) is *presupposition projection*. The full range of the behavior in various compound sentences exhibits some rather interesting patterns. There is limited consensus among many linguists about the basic pattern.

Here is one of the standard views of the presuppositions of complex sentences, based loosely on Karttunen (1973): Let A be any sentence whose presupposition is \underline{a} , and b be a sentence without any presupposition (for convenience I'll use lowercase letters to represent presupposition-less sentences, and capital letters to represent sentences with presuppositions). Then the following generalizations can be made.

1. $\neg A$ presupposes \underline{a}
2. $A \wedge b$ presupposes \underline{a}
3. $A \vee b$ presupposes \underline{a}
4. $A \rightarrow b$ presupposes \underline{a}
5. $b \wedge A$ presupposes $b \rightarrow \underline{a}$
6. $b \vee A$ presupposes $\neg b \rightarrow \underline{a}$
7. $b \rightarrow A$ presupposes $b \rightarrow \underline{a}$

These are rough-and-ready rules as they only define the presuppositions of sentences with a single binary operator in which just one of the component expressions has a presupposition. Note the asymmetry: $A \wedge b$ presupposes \underline{a} while $b \wedge A$ presupposes $b \rightarrow \underline{a}$. Given that conjunction is commutative, this might seem a bit surprising. The linguistic evidence for positing this asymmetry is complicated and something I'll return to later. Another thing to note is that the presuppositions predicted by rules 5. to 7. are conditionals. These rules explain why the following sentences fail to presuppose anything:

(4) John used to smoke and he's stopped.

- (5) John didn't use to smoke, or he's stopped
- (6) If John used to smoke, then he's stopped.

The presuppositions of sentences (4) to (6), given the rules 5. to 7. , are trivial. For instance, the presupposition of (4) by rule 5 is *if John used to smoke, then he used to smoke*. So the entire sentences are correctly predicted not to presuppose anything. The rules above can easily be elaborated into rules that predict the full presupposition of any complex sentence based on the presuppositions of its parts. Such a set of rules would be essentially the *filtering* rules from developed by Karttunen (1973). There is some debate over the empirical merits of these rules, but I want to put this aside here.³

Suppose rules along the lines of 1. to 7. suffice to describe the pattern of presupposition projection. They would still fail to *explain* in any way why the pattern of presuppositions project can be so described. Heim (1983) was a landmark paper because it gave a semantics of presuppositional expressions (and complexes formed out of these) that predict these rules of presupposition projection. I will outline her account and discuss the primary criticism that was leveled against it. Before I turn to this however, I need to discuss, in the next section, the Stalnakerian framework of common grounds used to analyze presuppositions.

3 Methodological Note: Common Grounds

One of the marks of linguistic presuppositions is that when a sentence presupposes another sentence an assertion of the sentence seems to take the other for granted. We thus might describe presuppositions by saying that a sentence P presupposes another sentence \underline{p} when an assertion of P is only felicitous in a context in which the mutual assumptions of the conversation participants include \underline{p} . This framework, borrowed from Stalnaker (1974), takes linguistic presupposition to give rise to acceptability conditions on the *common ground*, the collection of mutually accepted assumptions between conversational participants. My use of this framework is mostly for convenience: you might think presuppositions are instead something like default inferences, in which case you can rephrase everything in some other way.

³There is a long tradition that argues against these conditional presuppositions (most notably Geurts, 1996).

Here is a more careful description of the framework: in a conversation any utterance is made against a set of possible worlds, c , which is the set of worlds which are not ruled about by the mutual assumptions of the conversational participants. When one asserts a proposition, a , the normal effect, if the audience accepts the assertion, is the removal of the worlds where a is false from the common ground. One way of working presuppositions into this framework is to assume that certain sentences, such as A , are such that they are only felicitously asserted in certain common grounds. In particular, we say that if A presupposes \underline{a} , then A is only felicitously uttered in common grounds that entail \underline{a} (i.e. \underline{a} is true in every world in the common ground). When it is felicitous, the effect of an assertion of A is to remove certain worlds from the common ground.

We can restate the projection rules from Karttunen in terms of conditions on the common ground. (We will use the \models symbol to indicate that a set of possible world is such that some proposition is true in every world in it.)

1. $\neg A$ is acceptable in c iff $c \models \underline{a}$
2. $A \wedge b$ is acceptable in c iff $c \models \underline{a}$
3. $A \vee b$ is acceptable in c iff $c \models \underline{a}$
4. $A \rightarrow b$ is acceptable in c iff $c \models \underline{a}$
5. $b \wedge A$ is acceptable in c iff $c \models b \rightarrow \underline{a}$
6. $b \vee A$ is acceptable in c iff $c \models \neg b \rightarrow \underline{a}$
7. $b \rightarrow A$ is acceptable in c iff $c \models b \rightarrow \underline{a}$

This framework for describing presupposition is used in Heim's dynamic semantics. For our purposes, it will be useful to make one addition to/modification of the framework: If we have a sufficiently expressive language, we can also think of common grounds as being associated with *sentences* that are true only in the worlds in the common ground. So, I'll be discussing expressive languages such that for every set of possible worlds c , there is an atomic sentence s that is true in a world w iff $w \in c$. From now on when I talk about a common ground, c , I will sometimes refer to such a characteristic atomic sentence rather than the set of possible worlds itself. So, for

instance, we can understand $c \wedge a$ as meaning the conjunction of a sentence true only in worlds in c with the sentence a .

4 Heim’s Dynamic Semantics

This paper will formulate the issues and debate roughly the way it is in Heim (1983), rather than following the more extensive tradition that stems from Kamp (1981). This choice is both for convenience and because Heim’s work is more clearly directed at linguistic presuppositions rather than just the treatment of variables (though, of course, the dynamic tradition effectively assimilates these two issues). A crucial feature of Heim (1983) is that instead of assigning truth conditions (propositions) to sentences, she assigns rules for updating the common ground that may be undefined for some common grounds. Each atomic sentence A is associated with a means of updating the common ground, c , but the rule is only defined when c entails the encoded presupposition of A , \underline{a} . The effect of A when c is defined, in the propositional case, is equivalent to intersection with some proposition, which I’ll call the *force of A* , a' . To give a concrete example: if A is “John stopped smoking”, then \underline{a} will be the proposition expressed by “John used to smoke”. So A is only defined in common grounds which entail that John used to smoke. When A is defined the effect on a common ground, c , of asserting A will be to remove all those worlds from c in which John smokes now. So, the force of A , a' , is the proposition “John doesn’t smoke now.”⁴ We will write the result of updating the common ground c with A as $[A]c$.⁵ The key thing is that $[A]c$ (unlike typical well-formed sentences in logic) is sometimes undefined. When defined, of course, $[A]c$ is going to be true in a world if and only if $c \wedge a'$ is.⁶

Heim’s context change potentials for complex sentences are determined by the context change potentials of their parts in a compositional manner. For example, any sentence of the form $A \wedge B$ (where A and B are any CCPs) has the CCP given by what I will call a *procedure*. In the case of conjunction, $[A \wedge B]c$ is, by definition, equivalent to $[B][A]c$, which we can call the procedure for conjunction. What $[B][A]c$ means is that first we apply the procedure, A , to the common ground

⁴In fact, there will be many possible forces that do the work, since, for instance, “John doesn’t smoke now and John used to smoke” will have the same effect when on c when A is defined for c .

⁵Heim writes this as “ $c + A$.”

⁶We can also associate CCPs with non-presuppositional sentences. If a is proposition then we can define the CCP A s.t. \underline{a} is a tautology and $a' = a$. $[A]c$ then always equals $c \wedge a$, which is the normal state of affairs.

c , and then we take the resulting proposition and apply B to that, the result is what we take $A \wedge B$ to do to c (which we write $[A \wedge B]c$). We will assume, as Heim does, that if any part of a procedure is undefined the entire procedure is. If A and B are atomic CCPs with presuppositions \underline{a} and \underline{b} then it follows that the procedure $A \wedge B$ is defined just in case $c \models \underline{a}$ and $[A]c \models \underline{b}$. This matches the Karttunen rules precisely: if a has no presupposition then for $a \wedge B$ to be acceptable in Heim’s system it needs to be the case that $c \models a \rightarrow \underline{b}$.

It turns out that there are procedures for all logical connectives that predict Karttunen’s rules for presupposition projection based on their definedness conditions. Here are some standard procedures, taken from Heim and others:

$$[\neg A]c = c \wedge [\neg A]c$$

$$[A \vee B]c = [A]c \vee [B][\neg A]c$$

$$[A \rightarrow B]c = [\neg A]c \vee [B][A]c$$

So this story goes beyond the mere descriptive to give an actual semantics for both atomic and complex CCPs which predicts the basic pattern of presupposition projection.

5 The Standard Criticism

A serious problem with Heim’s approach is that different and, in some sense, equally good, procedures for some of the logical operators lead to different presuppositions. For example $[A][B]c$, captures conjunctions of the form $A \wedge B$, logically speaking, as well as $[B][A]c$ does, but it makes different predictions about the behavior of presuppositions under conjunction. In particular, if $[A \wedge B]c$ were equivalent to $[A][B]c$, then for $[A \wedge B]c$ to be acceptable it would need to be the case that $c \models \underline{b}$ and that $c + B \models \underline{a}$. On these rules the (7) would be infelicitous in any common ground that did not entail that John has a son—an obviously false prediction.

(7) John has a son and Bill knows it.

Thus, Heim’s dynamic semantic is not fully *explanatory* since it needs to stipulate a particular procedure for each connective, rather than allowing the procedure to follow from the truth-table

meaning of the connective.⁷

While in the case of conjunction, Heim's procedure is perhaps more *natural* it is not true, in the general case, that the most natural procedure yields the best predictions of the facts about presupposition projection. In fact, arguably, the most natural procedures predict almost none of the the right facts. For instance, the following are very simple, acceptable procedures for disjunction and conjunction that show the parallels between them:

$$[A \wedge B]c = [A]c \wedge [B]c$$

$$[A \vee B]c = [A]c \vee [B]c$$

However, these rules make very bad predictions about presupposition projections, as they simply predict universal inheritance of presuppositions from inside embeddings. So, trying to preserve the natural symmetries between the dual operators, \wedge and \vee , in the procedures for these operators, can lead to the wrong predictions. I think we ought to conclude that to the degree that Heim's dynamic semantics predicts the correct inheritance properties for presuppositions, these predictions depend on what can only be considered stipulations about the exact procedures for each of the logical operators. All else equal, this is undesirable.

Since this point often leads to some confusion, let me clarify a bit: I am not arguing that Heim's semantics for CCPs does not, in some sense, *explain* the Karttunen projection rules. The Karttunen rules do fall out from the definedness conditions for simple CCPs as well as the composition rules of the semantics. So Heim's account does have a significant explanatory advantage over the mere stipulation of Karttunen's rules. It is just that one might want to go further than this: one might want to somehow explain why, given a connective's truth-functional properties, it has the particular presupposition projection pattern it does. There is no a priori reason why we should think such an explanation is possible, but many theorists have the intuition that such an explanation is possible and some recent theories of presupposition projection do seem to achieve such explanatory power (Schlenker, 2006, to appear; Chemla, 2008; George, 2008).

⁷This criticism is well-known in the literature from Soames (1989) and Heim (1990, where the objection is attributed to Mats Rooth). Schlenker (2006, 2008) discuss the criticism in much detail.

6 A Non-Stipulative Treatment of the Complex Expressions

I will present a semantics for presuppositional expressions very similar to Heim's that does not depend on any stipulations peculiar to particular connectives (that are not predictable from their truth-conditions). The basic idea is that rather than stipulating procedures or templates for complex CCPs, we allow any procedure that is defined in a given context to be used. To realize this idea we need a language with two things: a syntactic specification of what counts as a procedure generally (which we will try to make as loose as possible) and a definition of when a given procedure is acceptable for a given connective.

To do this we will define a language L_1 that includes not only atomic sentences for propositions but also a class of CCPs. The syntax is much like Heim's language (as described above—her own syntax is slightly different): There is a set of atomic CCPs, which we will use uppercase letters, A, B, \dots . Any boolean combination of atomic CCPs is also considered an acceptable CCP. We also need to represent the common grounds (and updates of these) in this semantics and so we will have a class of lowercase letters, p, q, \dots representing atomic *sentences*. We also allow as a sentence the result of applying any CCP, X , to a sentence, p , which we write $[X]p$. In addition, we allow the usual boolean combinations of sentences.

The semantics of this language, for all but the case of complex CCPs, is exactly like Heim's dynamic semantics. An interpretation of the language assigns to each atomic sentence a pairing of worlds and truth values (so they are of type $\langle s, t \rangle$). As I mentioned earlier, in order to represent common grounds as atomic sentences, we will assume that for any set of possible worlds there is some atomic sentence that is true only in those worlds. An interpretation assigns to an atomic CCP, A , two sentences: the presupposition \underline{a} , the force, a' . For an atomic CCP, A , and a (defined) sentence p , $[A]p$ is defined iff p entails \underline{a} . If $[A]p$ is defined then it is true in a world w if and only if $p \wedge a'$ is true in w , as in Heim's semantics. (So atomic CCPs, as well as complex ones, are of type $\langle \langle s, t \rangle, \langle s, t \rangle \rangle$.)

So far, along with the usual assumptions of propositional language, this gives us a complete semantics for any sentence in L_1 that does not have a complex CCP in it. Before we give the

semantics for complex CCPs we need to introduce some new terminology. First we will define what it is for some sentence in L_1 to be a *procedure* for two CCPs and a sentence.

Syntactic Form of Procedures A sentence in L is a procedure for sentence p , and CCPs A and B iff it is a procedure according to these recursive rules:

1. p is a procedure for A, B , and p
2. if X and Y are procedures for A, B , and, p , then so are $X \wedge Y$, $X \vee Y$, and, $\neg X$
3. if X is a procedure for A, B , and p , then so are $[A]X$ and $[B]X$

Note that the procedure used for calculating $[A \wedge B]p$ on Heim's semantics, $[B][A]p$, is a procedure according to this definition, but so are some procedures for conjunction that don't work at predicting presuppositions, such as $[A][B]p$. We will say that a procedure for A, B , and p , is defined if and only if p is defined and every instance of $[A]$ or $[B]$ in it is defined.

The notion of a procedure picks out a very broad class of sentences. What we need now is a definition of what procedures suffice for capturing the semantics of a given boolean combination of CCPs:

Connectives and Procedures If u is a procedure for p , X , and Y , then u is acceptable for a binary connective $*$ iff, for all worlds w , sentences p' x' and y' , and atomic CCPs X' and Y' s.t. the force of X' and Y' is x' and y' , respectively, and the presuppositions of Y' and X' are both sentences equivalent to \top , $u_{p/p', X/X', Y/Y'}$ is true in w iff $p' \wedge (x' * y')$ is true in w (where $u_{p/p', X/X', Y/Y'}$ is the sentence formed by replacing every instance of p in u with p' , every instance of X by X' and every instance of Y by Y').

The idea here is the procedure must, when defined, be equivalent to what happens when we update by complex propositions in a bivalent semantics.

Instead of Heim's single procedure for each binary formula $A * B$, we now have an infinite set of acceptable procedures which are equivalent, in the bivalent case, to conjoining the common ground with $A * B$. We will say that $[A * B]c$ is defined iff there is some procedure for A , B , and c , acceptable for $*$ that is defined. If defined, $[A * B]c$ is true in w iff any defined procedure for A , B , and C that is acceptable for $*$ is true in w . This gives us a recursive semantics for complex

CCPs, since we now have a rule for determining the meaning of any complex CCP based on the meaning of its parts.

The theory may seem a bit cumbersome, so perhaps it is best to explain procedurally how it works. Essentially we are assuming that when faced with a complex CCP the audience can use any procedure that is defined to determine its meaning. This idea accords with the proposal for symmetric disjunction given in Soames (1982) where he suggests that there are different dynamic processes for understanding disjunction, and if one such procedure works without leading to a presupposition failure that is sufficient for the sentence to be acceptable. The technical implementation of this intuitive idea is, of course, more complex than Heim’s stipulative semantics, but that does not mean the underlying system is any less natural.⁸

This system yields specific predictions about how complex expressions inherit the presuppositions of their parts. The proofs of these predictions are in the appendix since they are somewhat complicated (given the freedom in the language the semantic properties are not so easily determinable). Here is a description of the inheritance rules, for negation (which is treated in a slightly different manner than two-place connectives) and disjunction and conjunction.⁹

- $[\neg A]c$ is defined iff $[A]c$ is defined.
- $[A \wedge B]c$ is defined iff $[A][B]c$ or $[B][A]c$ is defined.
- $[A \vee B]c$ is defined iff $[B][\neg A]c$ or $[A][\neg B]c$ is defined.

7 Making it Incremental

This system, of course, does not predict Heim’s (and Karttunen’s) asymmetric rules for presupposition projection. In other words, this system pays no attention to the order of commutative operators such as \vee and \wedge , whereas Heim’s system gives rules that differ for $A \wedge B$ and $B \wedge A$.

⁸I suppose, some might claim that the system here is not, in any legitimate sense, *dynamic*. People are entitled to use the term dynamic in a variety of ways, and I am sure the present system does not count as dynamic in many senses of the term in currency. However, I think that by being based on the idea that sentences are meaningful only if *some* dynamic procedure works, this form of semantics fits squarely into the dynamic program.

⁹I am omitting discussion of conditionals from the remainder of this paper. The predictions I make are perfectly normal, but I think given that conditionals are quite obviously not normal, binary truth-conditional operators, it is futile to treat them as if they were.

Recent work by Schlenker (2006, to appear) has suggested, however, a very simply and elegant way to transform a symmetric theory like this into an asymmetric one, like Heim's.

The basic idea is essentially a processing one. As we process a sentence from left-to-right we make sure that no matter how the sentence ends there will be no unsatisfied presuppositions in it. Following Schlenker, I will call a version of the theory with this feature an *incremental* version.

To make the theory above incremental we say that a complex CCP S is *incrementally acceptable* in c iff for any starting string of S , α , and any string β such that a) the only atomic CCPs in β are such that they are always defined and b) $\alpha\beta$, the concatenation of α and β , is a well-formed complex CCP, $[\alpha\beta]c$ is defined. It turns out that this system is exactly equivalent to Heim's asymmetric system as described above. The proof of this last point, which proceeds by induction, is in Section 11.

8 Capturing Symmetries in Presupposition Projection

Despite a preference in the literature for asymmetric theories of presupposition projection, there are many cases which can only be handled by a symmetric theory. Usually the cases are slightly more complex than the very standard cases, but I think the judgments are relatively clear.¹⁰

The following sentences are examples where the standard asymmetric theories predict that there are presuppositions, but the non-incremental version of the theory above predicts no presuppositions:

¹⁰These observations build on work by Schlenker (to appear, 2008). The reason why we need to look at complex cases is there may be independent pragmatic principles interfering with our judgments in many simple cases. For example, the reason $A \wedge \underline{a}$ is unacceptable may be that there is a prohibition against saying $A \wedge B$ if A entails B (but not vice versa). So, for example, as Schlenker notes, the following sort of sentence is odd:

- (a) John is a practicing, accredited doctor and he has a medical degree.

Whereas the reverse order is more normal:

- (b) John has a medical degree and he is a practicing, accredited doctor.

Schlenker thinks such pragmatic facts are themselves part of what explain presupposition projection—and he gives an account that builds on such things as asymmetric redundancy rules. But the existence of symmetric presupposition phenomenon casts some doubt on this claim. Nonetheless, Schlenker argues that there are persistent asymmetries in presupposition projection that cannot be explained by simple constraints on entailing conjuncts and the like.

A	$\neg A$
T	F
F	T
*	*

$A \vee B$	T	F	*
T	T	T	T
F	T	F	*
*	T	*	*

$A \wedge B$	T	F	*
T	T	F	*
F	F	F	F
*	*	F	*

Figure 1: Strong-Kleene Truth Tables

- (8) If John doesn't know it's raining and it is raining heavily, then John will be surprised when he walks outside.
- (9) It's unlikely that John still smokes, but he used to smoke a lot.
- (10) Either the bathroom is well hidden or there is no bathroom.

In all these cases, I find the standard judgement is that there is no presupposition perceived nor do the examples seem to be marked in a way that indicates cancellation. Given examples like this we might want to keep both the symmetric and the asymmetric versions of the theory, as Schlenker suggests for his Transparency Theory.¹¹

9 Strong-Kleene and the Looser Definition of Procedure

One classic symmetric theory of presupposition projection worth comparing the present account to is one based on the strong-Kleene truth-tables. These trivalent truth-tables are given in Figure 1.

We could use these tables to yield a system of presupposition projection in a quite straightforward way: Suppose a presuppositional sentences A is true or false in w iff a is true in w . We then

¹¹I discuss the issue of symmetries in some more depth in Rothschild (forthcoming).

assume that we can utter a proposition, A , in a common ground, c , iff A is true or false in every world in c .¹² As George (2008) observes, if you add an asymmetry to this system by checking incrementally, you will derive the standard Karttunen-Heim projection rules.

While the asymmetric versions of the dynamic account above and the strong-Kleene theory are equivalent, the symmetric versions are not. This can be seen by looking at a sentence like $(A \wedge \underline{b}) \wedge (B \wedge \underline{a})$. In the dynamic theory given above this is not acceptable if c does not entail either $\underline{b} \rightarrow \underline{a}$ or $\underline{a} \rightarrow \underline{b}$, however, in a strong-Kleene system this sentence is true or false in every possible world, and so admissible in any c .

I show in Section 11, however, that when you loosen the constraints on what counts as a procedure, you can get a dynamic system that yields exactly the same results as a strong-Kleene system. The particular constraint that is changed is that we now allow any arbitrary sentence to be part of an update procedure. This result may seem surprising to those who have thought of Heim's dynamic semantics as being radically different from a simple trivalent account. But, treating sentences as partially defined functions—as long as they are well-behaved functions—turns out to be very similar to treating sentences as propositions with truth-value gaps.

10 Quantification/Donkey Anaphora

When we expand our language to include quantifiers, variables, and presuppositional predicates new problems arise. We want to be able to predict the presuppositions of sentences such as (11).

(11) Every doctor knows that he should quit smoking.

To handle cases like this, we need to have CCPs with variables in them and a semantics that gives the meaning and definedness conditions of complex CCPs involving quantifiers. This can be done in a variety of ways and the choice involved will have some effect on the predictions about presupposition projection under quantification. Heim's treatment of quantifiers, which would take us a bit far afield to review here, essentially stipulated that the matrix (or scope) position had a different treatment than the restrictor position.

¹²See Soames (1989) for discussion of why this principle is not actually as plausible as it seems.

The formal techniques employed here can certainly be expanded to a quantification fragment. The predictions yielded, will depend on the particular definitions one gives for the syntactic form of “procedures” for quantifiers. For the purposes of this paper I have decided not to include a quantificational fragment since it would significantly complicate the already lengthy presentation of the language in Section 11 and the basic principles needed are not novel. Also it is not clear how to judge theories for quantified presuppositions since it is hard to know what the actual observed data is for presupposition projection under quantifiers.¹³

Another issue that arises in this context is how this version of dynamic semantics might be extended to treat donkey anaphora. I suspect the framework could be expanded to yield a natural treatment of donkey anaphora along the lines of Heim (1982) and Kamp (1981). However, I do not explore this question here.

11 The Language

11.1 Syntax

We will describe a language L_1 . The syntax is the standard syntax of propositional logic with an infinite set of CCPS (which are akin to modalities). I will use lowercase letters to represent sentences and uppercase letters to represent CCPS:

- p, q, \dots are atomic sentences
- if p is a sentence and q is a sentence then so are $p \wedge q, p \vee q, \neg p$.
- A, B, \dots are atomic CCPs.
- if X is a CCP and Y is a CCP then so are $X \wedge Y, X \vee Y, \neg X$.
- if p is a sentence and X is a CCP, then $[X]p$ is a sentence.

¹³Chemla (2007) reports experiments that indicate differences between different quantifiers in how they project presuppositions.

11.2 Semantics I

We will give a semantics in terms of a primitive set of possible worlds W . The interpretation I assigns sentences and worlds to truth-values, so for each atomic sentence p either $I(w)(p) = 1$ or $I(w)(p) = 0$, if $I(w)(p) = 1$ we write $w \models p$ otherwise $w \not\models p$. And for any sentence s , $w \models s$ will be used to indicate that s is true in w on I . For convenience we will make an expressivity assumption:

Expressivity For every subset of W , u , there is some atomic sentence p such that $w \models p$ iff $w \in u$.

We can now give recursive semantics for the standard propositional logic part of this language:

Basic Sentential Rules For any sentences a and b :

- $w \models a \wedge b$ iff $w \models a$ and $w \models b$
- $w \models a \vee b$ iff $w \models a$ or $w \models b$
- $w \models \neg a$ iff $w \not\models a$

We can think of \rightarrow as a shorthand in the usual way.

We also need to give the semantics of $[X]p$ where X is a CCP and p is a sentence. The interpretation I assigns to each atomic CCP, X , two atomic sentences of propositional logic, its force x' , and its presupposition, \underline{x} .

Now for an atomic CCP X : $[X]p$ is defined if and only if for every w : $w \models p \rightarrow \underline{x}$ (from now on we will represent this condition by $p \models x$). If $X[p]$ is defined then $w \models X[p]$ iff $w \models p \wedge x'$. As a second expressivity assumption, we assume that for every pair of atomic sentences x and y there is some CCP X such that x is the presupposition of X and y is the force of X .

A given sentence is *defined* iff every subsentence it of the form $[X]p$ where X is a CCP and p is a sentence, is defined. Sentences that are not defined are neither true nor false in any world.

It is now left to to give the semantics for complex CCPs which we will do in section 11.4.

Note. We can think of (atomic or complex) sentences as common grounds since they pick out a set of worlds in which they are true or false. We can think of $[X]p$ as representing the result of

updating the common ground p with the CCP X where there is no presupposition failure if and only if $[X]p$ is defined.

11.3 Interlude

For any sentence p and CCPs, X and Y , a sentence u is a *procedure* for p , X , and Y iff it can be constructed with the following recursive rules:

- p is a procedure for p , X , and Y
- if s and t are procedures for p , X , and Y then $\neg s$, $s \vee t$, and $s \wedge t$ are procedures for p , X , and Y .
- if s is a procedure for p , X , and Y then $[X]s$ and $[Y]s$ are procedures for p , X , and Y

We will now give a general definition of when a procedure, u , is acceptable for a given connective. If u is a procedure for p , X , and Y , then u is acceptable for a binary connective $*$ iff, for all worlds w , atomic sentences p' , x' , and y' , and atomic CCPs X' and Y' s.t. the force of X' and Y' is x' and y' , respectively, and the presuppositions of Y' and X' are both sentences equivalent to \top , $w \models u_{p/p', X/X', Y/Y'}$ iff $w \models p' \wedge (x' * y')$ (where $u_{p/p', X/X', Y/Y'}$ is the sentence formed by replacing every instance of p in u with p' , every instance of X by X' and every instance of Y by Y').

Note: the expressivity assumptions above give this requirement its force. It follows immediately that if a procedure u for an atomic sentence p and atomic CCPs X and Y acceptable for $*$ is defined, then it is true if and only if $p \wedge (x' * y')$ is true.

(We define procedures for a sentence p and a single CCP X and their acceptability for the operator \neg in an analogous manner, limiting the CCPs in the procedure to just X .)

11.4 Semantics II

It remains to define the effect of complex CCPs on sentences. For any sentence p , CCPs X and Y , and binary operator, $*$, $[X * Y]p$ is defined if and only if there exists some procedure, u , for p , X , and Y acceptable for $*$ s.t. u is defined. If $[X * Y]p$ is defined then (for all w) $w \models [X * Y]p$ iff there exists a procedure u for p , X and Y , s.t. u is defined, $w \models u$. (*Mutatis mutandis* for $[\neg X]p$.)

This defines complex CCPs in terms of simpler ones and so it completes our recursive semantics for the language.

11.5 Properties of the Language

Proposition 1. *If there are CCPs X and Y and sentences x' and y' , s.t. for any p such that $[X]p$ is defined, (for all w) $w \models [X]p$ if and only if $w \models p \wedge x'$ and for any p such that $[Y]p$ is defined, for all w , $w \models [Y]p$ iff $w \models p \wedge y'$, then for any p such that $[X * Y]p$ is defined, for all w , $w \models [X * Y]p$ if and only if $w \models p \wedge (x' * y')$.*

Proof. This is immediate in the case where X and Y are atomic CCPs with forces x' and y' , by the definition of procedure acceptable for $*$. For non-atomic CCPs, X and Y , by assumption, they act exactly like atomic CCPs with force x' and y' when defined, then this also holds for them. \square

Since for an atomic CCP, X_i , $[X_i]p$, when defined, is equivalent to $p \wedge x'_i$ then, for any non-atomic CCP, X , let us call x' the sentence formed by replacing each atomic CCP $X_i \in X$ with x'_i . It follows from fact 1, by induction, that if $[X]p$ is defined then $w \models [X]p$ iff $w \models p \wedge x'$. We can call x' the force of X .

Let us call any complex CCP, X , *well-behaved* iff: if $[X]p$ is undefined then for any q s.t. $p \models q$, $[X]q$ is undefined. We will prove various facts about well-behaved CCPs and then show that all CCPs are well-behaved.

Proposition 2. *For any well-behaved CCPs, X and Y , $[X \wedge Y]p$ is defined if and only if $[X][Y]p$ or $[Y][X]p$ is defined.*

We can ignore the case where p is undefined, since then for any CCP, Z , $[Z]p$ is undefined.

The right-to-left direction is immediate, since each sentence on the right-hand side corresponds to a procedure for p , X , and Y acceptable for conjunction, and so if one of them is defined then the complex CCP is defined. So we need to show that $[X \wedge Y]p$ is defined only if $[X][Y]p$ or $[Y][X]P$ is defined. We will do this in two steps:

Proposition 3. *$[X \wedge Y]p$ is defined only if $[Y]p$ or $[X]p$ is defined.*

Proof. Suppose, for a contradiction that neither $[Y]p$ nor $[X]p$ is defined. We will show that every procedure u for p , X , and Y that is defined is such that $p \models u$ or $u \models \neg p$. We will prove this by induction on the construction of procedures. For the base case, $u = p$, obviously $p \models p$. We will, first, show that the disjunctive induction property is preserved under disjunction, conjunction, and negation (i.e. Boolean combination). For negation, suppose $p \models u$ then $\neg u \models \neg p$, and if $u \models \neg p$ then $p \models \neg u$. For conjunction, suppose $p \models u$ and $v \models \neg p$ then $u \wedge v \models \neg p$. For disjunction suppose $p \models u$ and $v \models \neg p$ then $p \models u \vee v$. Now the question is whether there is any defined instance of adding a CCP which removes the property. Consider some u s.t. $p \models u$. By assumption $[X]p$ and $[Y]p$ are undefined, so since $[X]$ and $[Y]$ are well-behaved, $[X]u$ and $[Y]u$ are undefined. On the other hand, if $u \models \neg p$ then $[X]u$ or $[Y]u$, if they are defined, will entail $\neg p$.

Note that the proof that any of the defined procedure for p , X , Y , satisfies the property did not depend on any features of p , X or Y (though the fact that certain procedures were undefined did). It follows that for any atomic sentence p' and atomic CCPs X' and Y' and any defined procedure u for p , X , and Y , $u_{p/p', X/X', Y/Y'} \models \neg p'$ or $p' \models u_{p/p', X/X', Y/Y'}$. This makes it the case that u cannot be acceptable for conjunction. □

Proposition 4. *If $[Y]p$ is not defined then, $[X \wedge Y]p$ is defined only if $[Y][X]p$ is defined.*

Proof. We prove this in a similar manner to the previous proposition.

We know from the previous proposition that if $[Y]p$ is defined then $[X]p$ must be defined if $[X \wedge Y]p$ is defined, so let us suppose that $[X]p$ is defined. For a contradiction suppose that $[Y][X]p$ is not defined.

Consider any procedure u for p , X , and Y . By induction we will show that u is such that, if defined, either $[X]p \models u$ or $u \models \neg[X]p$. The only atomic sentence is p . $[X]p \models p$ and $\neg p \models \neg[X]p$. By the same arguments as in Proposition 3 this property is preserved for the boolean combinations of sentence with this property. It is also clearly preserved by applying the CCP $[X]$. For the case of $[Y]$: If $[X]p \models u$ then $[Y]u$ will be undefined, since $[Y]$ is well behaved, and if $u \models \neg[X]p$ then so will $[Y]u$.

It follows that no defined u for p , X , and Y , is an acceptable procedure for conjunction. □

By symmetry this concludes our proof of Proposition 2. □

Proposition 5. *For any well-behaved CCPs, X , $[\neg X]p$ is defined if and only if $[X]p$ is defined*

Proof. The right-to-left direction is trivial. Suppose $[X]p$ is undefined. Then every defined procedure u for p and X will be such that either $p \models u$ or $p \models \neg u$. It follows that no such u will be acceptable for negation. \square

Proposition 6. *For any well-behaved CCPs, $[X \vee Y]p$ is defined if and only if $[X][\neg Y]p$ or $[Y][\neg X]p$ is defined.*

Proof. The right-to-left direction is obvious since, if either condition is met, then procedures acceptable for disjunction that are defined are easily found.

For the left-to-right direction, we can proceed in two steps. By the exact argument of proposition 3 it follows that if $[X \vee Y]p$ is defined then $[Y]p$ or $[X]p$ is defined.

By symmetry it will suffice to show that if $[X]p$ is defined, then $[X \vee Y]p$ is defined only if $[Y][\neg X]p$ is defined.

We will show that any procedure, u , for p , X , and Y is such that either $[\neg X]p \models u$ or $u \models \neg[\neg X]p$. Obviously, $[\neg X]p \models p$. This property again is preserved under boolean combination. Suppose $[\neg X]p \models u$ then $[X]u \models \neg[\neg X]p$, and if $u \models \neg[\neg X]p$, then $[X]u \models \neg[\neg X]p$. If $[\neg X]p \models u$ then $[Y]p$ is undefined, if $u \models \neg[\neg X]p$ then $[Y]u \models \neg[\neg X]p$.

Since any procedure u for p , X , and Y is such that either $[\neg X]p \models u$ or $u \models \neg[\neg X]p$ (regardless of what p is or the force of X and Y) no such u can be acceptable for \vee . \square

Proposition 7. *Every CCP is well-behaved.*

We can prove this by induction on the complexity of the CCP. The base case is trivial. Now on the complexity of the CCPs, if X and Y are well-behaved, then $[X \wedge Y]p$ is defined only if $[X][Y]p$ or $[Y][X]p$ is defined. From this it follows that for all p and q such that $p \models q$, when $[X \wedge Y]q$ is defined so is $[X \wedge Y]p$. So $[X \wedge Y]$ is well behaved. Similarly for negation and disjunction.

11.6 Looser semantics

We will defined a language L_2 that differs from L_1 only in having a different definition of a *procedure*. The difference is only that any atomic sentence is now allowed in the construction of a procedure for a sentence p and CCPs, X and Y . Here is the new recursive definition of procedure:

- any atomic sentence s is a procedure for p , X , and Y
- p is a procedure for p , X , and Y
- if s and t are procedures for p , X , and Y then $\neg s$, $s \vee t$, and $s \wedge t$ are procedures for p , X , and Y .
- if s is a procedure for p , X , and Y then $[X]s$ and $[Y]s$ are procedures for p , X , and Y

The semantics is otherwise the same as L_1 and may be determined by the same interpretation I .

11.7 Supervaluationist Equivalent

First, to facilitate our proofs, we will introduce a language with supervaluations but without CCPs, L_s , based on the supervaluationist language in Schlenker (2008). The syntax of this language is the standard one of propositional logic with two kinds of sentences, capital-letter ones and lowercase-letter ones (which differ semantically but not syntactically). This language has not just one interpretation but an infinite set of interpretations I_s which bears certain links to I .

Every interpretation in I_s agrees with I for all the lower case sentences in P_1 . An uppercase letter atomic supervaluationist sentence X (where I assigns the CCP X the force x' and the presupposition \underline{x}) is such that on every interpretation in I_s , $w \models X$ if $w \models \underline{x} \wedge x'$ and $w \not\models X$ if $w \models \underline{x} \wedge \neg x'$. Besides these restrictions we assume that I_s contains every possible interpretation. There are no CCPs in this language.

We say that $\text{super}(w \models s)$ if every interpretation in I_s makes this the case (we can also say that s is *supertrue* in w). We write $\text{super}(s)$ to indicate that, for all w , either $\text{super}(w \models s)$ or $\text{super}(w \models \neg s)$.

We will construct a mapping, f , from sentences in L_2 to sentences in L_s . For any sentence S in L_2 , $f(S)$ is found by doing the following two things: 1) replacing every instance of $[X]p$ (where p is any sentence and X is any CCP with $(X \wedge p)$, and 2) taking the resulting sentence in L_s and numbering each instance of any atomic CCP that appears more than once, from right-to-left. So, for instance, $f([X]p \vee [X \wedge Y]q) = (X_1 \wedge p) \vee ((X_2 \wedge Y) \wedge p)$.

What an interpretation in I_s assigns to X_i is subject to the same rules as it is for X (but these rules do not fully determine what each interpretation assigns to X_i). This means that for every w where $w \not\models \underline{x}$ there is some interpretation in I_s where $w \models X_1$ but $w \models \neg X_2$. So, while for every w and X , $w \models X \vee \neg X$, it is not the case that for every w , X_1 and X_2 $w \models X_1 \vee \neg X_2$ if \underline{x} .

Now for any given sentence S in L_2 we can talk about the sentence that f maps it to in L_s . We will do this without explicitly writing $f(S)$, but rather just using S , and making clear by context which sentence we are talking about.

Let us compare this system with a standard strong-Kleene semantics. On this semantics the interpretation function I_k assigns each atomic sentence a truth value in each world of T F or N we assume I_k agrees with I for all atomic sentences. For atomic CCP X , I_k assigns X T in all worlds where $\underline{x} \wedge \underline{x}'$ is true, F in all worlds where $\underline{x} \wedge \neg \underline{x}'$ is true, and N in all other worlds. The recursive semantics goes as follows:

- $\neg s$ is T in w iff s is F in w , $\neg s$ is F in w iff s is T in w , otherwise $\neg s$ is N in w .
- $s \wedge r$ is T in w if s and r are T in w , $s \wedge r$ is F in w if s or r is F in w , otherwise $s \wedge r$ is N in w .
- $s \vee r$ is T in w if s or r are T in w , $s \vee r$ is F in w if s and r is F in w , otherwise $s \vee r$ is N in w .

Schlenker (2008) proves that this relation holds between strong-Kleene systems and supervaluationist systems:

Proposition 8. *If no atomic sentences repeats in a given sentence s , then $\text{super}(w \models s)$ iff on the strong Kleene system s is T in w , and $\text{super}(w \models \neg s)$ iff on the strong Kleene system s is F in w .*

11.8 Definedness Conditions with Looser Semantics

Proposition 9. *For any CCP X and a sentence p , such that $\text{super}(p)$, the sentence $[X]p$ is defined iff $\text{super}([X]p)$.*

We will prove this proposition by induction over the complexity of CCPs. The base case is relatively simple:

Proposition 10. *For atomic CCP/supervaluationist sentence X , $\text{super}([X]p)$ iff $[X]p$ is defined.*

Proof. $[X]p$ is defined iff $p \models \underline{x}$. Now we need to show that for all w , $f([X]p)$ is super-true or super-false in w iff $p \models \underline{x}$. Suppose $p \not\models \underline{x}$. Then there exists a w such that p is true in w and \underline{x} is false. It follows that $f([X]p)$ is not true or false on all interpretations in I_s in this world, so it's not the case that $\text{super}([X]p)$. Now suppose $p \models \underline{x}$ then in each world w either \underline{x} is true in which case $f([X]p)$ must be super-true or super-false, or $w \models \neg p$ in which case $f([X]p)$ is super-false in that world. \square

We will say in general for the (complex) supervaluationist sentence X that the sentence \underline{x} is such that $\text{super}(w \models X)$ or $\text{super}(w \models \neg X)$ iff $w \models \underline{x}$. Given Proposition 10 this comports with our prior used of underlining for atomic sentences.

Proposition 11. *For all sentences p such that $\text{super}(p)$ and such that p is defined, and all complex CCPs X , if $[X]p$ is defined iff $\text{super}([X]p)$ then $[\neg X]p$ is defined iff $\text{super}([\neg X]p)$.*

Proof. We can assume that $[X]p$ is defined iff $\text{super}([X]p)$. It is obvious that $\text{super}([\neg X]p)$ is defined iff $\text{super}([X]p)$, so if $\text{super}([\neg X]p)$ is defined then so is $[X]p$, and thus $[\neg X]p$ is defined.

Suppose, that $[\neg X]p$ is defined in L_2 . Then there exists a u for p and X acceptable for \neg such that u is defined. Now we need to show that $\text{super}(u)$. Let us do this by induction on the complexity of u . The simplest sentence must be p or some other sentence without any CCPs so this is trivial. If $\text{super}(s)$ and $\text{super}(t)$ then any boolean of s and t will also be super-true or super-false in all worlds. So the only thing we need to look at is instances of the CCP X in u . Now we know that each instance of $[X]t$ in u is defined and thus by induction assumption, if $\text{super}(t)$, then $\text{super}([X]t)$. This completes the proof by induction that $\text{super}(u)$. By construction, if u is a procedure for p and X acceptable for \neg then u is logically equivalent to $[\neg X]p$ in L_s . So if $\text{super}(u)$ then $\text{super}([\neg X]p)$, since supervaluationist sentences have the same truth value on each interpretation across logical equivalencies. \square

Proposition 12. *Consider complex CCPs X and Y such that $[X]p$ is defined iff $\text{super}([X]p)$ and $[Y]p$ is defined iff $\text{super}([Y]p)$, the following to properties hold:*

- $[X \wedge Y]p$ is defined iff $\text{super}([X \wedge Y]p)$

- $[X \vee Y]p$ is defined iff $\text{super}([X \vee Y]p)$

Proof. First we need to show that if $[X \wedge Y]p$ is defined, then $\text{super}([X \wedge Y]p)$, and if $[X \vee Y]p$ is defined, then $\text{super}([X \vee Y]p)$. If $[X \wedge Y]p$ is defined then there is some procedure, u acceptable for p , X , and Y , and we can show as in Proposition 11 that $\text{super}(u)$, and, so, $\text{super}([X \vee Y]p)$.

Now we need to show that if $\text{super}([X \wedge Y]p)$ then $[X \wedge Y]p$ is defined. To do this consider the procedure u for p , X , and $Y =$ the conjunction of 1 to 4

$$[X][Y](p \wedge \neg \underline{x} \wedge \neg \underline{y}) \tag{1}$$

$$[X][Y](p \wedge \underline{x} \wedge \underline{y}) \tag{2}$$

$$[X][Y](p \wedge \neg \underline{x} \wedge \underline{y}) \tag{3}$$

$$[Y][X](p \wedge \underline{x} \wedge \neg \underline{y}) \tag{4}$$

It should be obvious that u is acceptable for conjunction.

Now the only question is whether u is defined. Here we will make use of Proposition 8 to determine the truth-conditional properties of u which are more transparent on a strong-Kleene system.

We can assume that $\text{super}([X \wedge Y]p)$. Given fact 8, this means that for all $w : w \not\models \neg \underline{x} \wedge \neg \underline{y}$. It follows that sentence 1 is defined, since for any CCP X , $[X]p$ is defined when p is false in all worlds.

Sentence 2 must be defined also since $[X]s$ is defined if $s \models \underline{x}$ and $[Y]s$ is defined if $s \models \underline{y}$.

For sentence 3. We know the instance of $[Y]$ in it is defined. If $\text{super}([X \wedge Y]p)$ then there can be no world w such that $y' \wedge \neg \underline{x} \wedge \underline{y}$, which means that the instance of $[X]$ must also be defined.

Sentence 4 must be defined by symmetry.

We can now construct a procedure for disjunction using those for conjunction and negation since $A \vee B$ is equivalent to $\neg(\neg A \wedge \neg B)$. □

This completes our proof of Proposition 9. □

11.9 Incremental Acceptability of CCPs

I consider here an incremental version of the semantics. We will say that a CCP S in L_1 is *incrementally acceptable* for c iff for every starting string α of S and every string β such that the concatenation of α and β , S' , is a well-formed CCP and for every atomic CCP X , $\underline{x} = \top$, $[S']c$ is defined.

We will compare this with what I'll call Heim's system. This is the same as L_1 except for the definitions for the complex CCPs are not any procedure but only as follows:

- $[\neg A]c = c \wedge \neg[A]c$
- $[A \wedge B]c = [B][A]c$
- $[A \vee B]c = [A]c \vee [B][\neg A]c$

Any complex CCP is defined if and only if these rules yield a sentence that is defined, otherwise it is undefined. It should be obvious that if some formula is defined on Heim's rules than it is defined in L_1 .

Proposition 13. *For all sentences c , S is incrementally acceptable in c iff S is Heim-acceptable in c .*

Proof. First suppose that S is incrementally acceptable in L_1 . We need to show that S is Heim acceptable. We will show this by induction on operators:

Base case: trivial.

Suppose for both A and B if they are incrementally acceptable then they are Heim-acceptable.

Suppose $\neg A$ is incrementally acceptable in c . Suppose A is not incrementally acceptable in c . Then there exists an A' which share a starting string with A and has only CCPs with trivial presuppositions afterwards such that $[A']c$ is not defined. It follows $[\neg A']c$ is not defined, which contradicts our assumption. So A is incrementally acceptable in c , which means that A is Heim-acceptable in c , and therefore $\neg A$ is also.

Suppose $A \wedge B$ is incrementally acceptable in c . If it is incrementally acceptable in c then $[A \wedge B]c$ is defined. If so, then either $[B][A]c$ is defined or $[A][B]c$ is defined. If it is only the latter then $A \wedge B$ is not incrementally acceptable in c . So $[B][A]c$ is defined. Now we need to

show that both $[B]$ is incrementally acceptable in $[A]c$, and A is incrementally acceptable in c . The latter follows immediately from the incremental acceptability of $A \wedge B$ in c . Suppose $[B]$ is not incrementally acceptable in $[A]c$, then there is a B' with an initial string identical to an initial string of B (and such that all the atomic CCPs after this string are always defined) is such that $[B']c$ is not defined. It follows that $[A \wedge B']c$ is not defined (since either $[B']c$ or $[B']c$ must be defined for $[A \wedge B']c$ to be defined). This conflicts with our assumption that $A \wedge B$ is incrementally acceptable. It follows that B is incrementally acceptable in $[A]c$, and thus $A \wedge B$ is Heim acceptable.

Suppose $A \vee B$ is incrementally acceptable in c . Then $[A \vee B]c$ is defined so either $[B][\neg A]c$ or $[A][\neg B]c$ is defined. It cannot only be the latter, so $[B][\neg A]c$ must be defined. We know that A is incrementally acceptable in c , and so A is Heim acceptable in c , and so $[\neg A]$ is Heim acceptable in c . It remains to show that $[B]$ is incrementally acceptable in $[\neg A]c$. Suppose it is not. Then there is some B' that shares an initial string with B (and such that all the atomic CCPs after this string are always defined) such that $[B']c$ is not defined. But in this case $A \vee B$ is not incrementally well defined in c . So $[B]$ is incrementally acceptable in $[\neg A]c$ and thus is Heim acceptable in $[\neg A]c$ and so $A \vee B$ is Heim acceptable in c .

Now suppose S is Heim acceptable, we need to show that it is also incrementally acceptable. By induction:

Base case: trivial.

Suppose both A and B are such that if they are Heim-acceptable they are incrementally acceptable.

Consider $\neg A$. If it is Heim acceptable in c then A is Heim acceptable in c , so A is incrementally acceptable in c , which means that $\neg A$ is incrementally acceptable.

Consider $A \wedge B$. If it is Heim-acceptable in c then A is Heim acceptable in c so A is incrementally acceptable in c . Likewise B is incrementally acceptable in $[A]c$, since it is Heim acceptable in $[A]c$. Suppose $A \wedge B$ is not incrementally acceptable in c . Given that A is incrementally acceptable this means some B' with which shares an initial string with B (and such that all the atomic CCPs after this string are always defined) such that $[A \wedge B']c$ is not defined. But since $[A]c$ is defined, this means that $[B']c$ is not defined, which contradicts the fact that B is incrementally acceptable

in $[A]c$. So $A \wedge B$ must be incrementally acceptable in c .

Suppose $A \vee B$ is Heim acceptable. It follows that A is Heim acceptable in c , and so A is incrementally acceptable in c . Likewise B is Heim acceptable in $[\neg A]c$ and so B is incrementally acceptable in $[\neg A]c$. Now, suppose $A \vee B$ is not incrementally acceptable in c . Then there exists some B' (and such that all the atomic CCPs after this string are always defined) with the same starting string as A such that $[A \vee B']c$ is not defined. But since $[A]c$ is defined, this means that $[B'][\neg A]c$ is not defined, which contradicts the fact that B' is incrementally acceptable in $[\neg A]c$. So $A \vee B$ is incrementally acceptable in c . \square

References

- Beaver, David (2001). *Presupposition and Assertion in Dynamic Semantics*. CSLI.
- Chemla, Emmanuel (2007). Presuppositions of quantified sentences: Experimental data. Manuscript, ENS.
- (2008). Similarity: Towards a unified account of scalar implicatures, free choice permission and presupposition projection. Unpublished manuscript, ENS.
- George, Benjamin (2008). A new case for an old logic: reviving strong Kleene approaches to presupposition projection. Unpublished manuscript, UCLA.
- Geurts, Bart (1996). Local satisfaction guaranteed: A presupposition theory and its problems. *Linguistics and Philosophy*, 19:259–294.
- Heim, Irene (1982). *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts, Amherst.
- (1983). On the projection problem for presuppositions. In D. Flickinger and M. Wescoat (eds.), *WCCFL 2: Second Annual West Coast Conference on Formal Linguistics*, pages 114–125. Blackwell.
- (1990). On the projection problem for presuppositions. In S. Davis (ed.), *Pragmatics*. Oxford University Press.

- Kamp, Hans (1981). A theory of truth and semantic representation. In J. Groenendijk et al. (ed.), *Formal Methods in the Study of Language*, pages 277–322. Mathematish Centrum.
- Karttunen, Lauri (1973). Presuppositions of compound sentences. *Linguistic Inquiry*, 4:169–93.
- Rothschild, Daniel (forthcoming). Presupposition projection and logical equivalence. *Philosophical Perspectives*.
- Schlenker, Philippe (2006). Anti-dynamics: Presupposition projection without dynamic semantics. *Journal of Logic, Language, and Information*, 16:325–56.
- (2008). Local contexts. Manuscript, Institut Jean-Nicod and NYU.
- (to appear). Be articulate: A pragmatic theory of presupposition projection. *Theoretical Linguistics*.
- Soames, Scott (1982). How presuppositions are inherited: A solution to the projection problem. *Linguistic Inquiry*, 13:483–545.
- (1989). Presuppositions. In D. Gabbay and F. Guenther (eds.), *Handbook of Philosophical Logic*, volume IV, pages 553–616. Dordrecht.
- Stalnaker, Robert (1974). Pragmatic presuppositions. In Milton K. Munitz and Peter K. Unger (eds.), *Semantics and Philosophy*. NYU.