

# Predicting Presupposition Projection: some alternatives in the strong Kleene tradition\*

Benjamin R. George  
brg42@ucla.edu

2 March 2008

## Abstract

This paper presents an architecture for evaluating presuppositional expressions that predicts the projection behavior of functions from their (bivalent) truth-conditional contributions. The resulting system demonstrates how a static, trivalent account can compete with other alternatives to address certain theoretical complaints raised against older projection theories, and makes a number of interesting new predictions.

## 1 Some Problems for a Theory of Presupposition Projection

This paper develops a semantic framework intended to overcome some of the weaknesses of the major theories of presupposition projection currently on the market, especially those associated with the dynamic semantics tradition. It seeks to avoid specifying projection behavior of functions in their lexical meanings, and to instead provide a general framework that derives projection facts from word meanings that are comparable in form and in content to traditional, static, bivalent meanings. In this respect, its goals

---

\*This manuscript owes a great deal to insights and suggestions from Philippe Schlenker, Emmanuel Chemla, Edward Keenan, Giorgio Magri, Roni Katzir, Danny Fox, Marcus Kracht, Edward Stabler, Daniel Büring, and all the participants in the Fall 2007 UCLA presupposition seminar. The work that produced it was supported in part by a graduate research mentorship for the 2007-2008 school year and a 2007 graduate summer research mentorship, provided by the UCLA graduate division, and by NSF grant BCS-0617316. I claim sole credit for its many shortcomings.

are similar to those of programs pursued of Schlenker [11], Chemla [2], Fox [3], LaCasse [7] and others. The present system’s core mechanics are based on intuitions underlying the strong Kleene [8] system and the trivalent truth functional approaches of Peters [10] and Kracht [9], and draw on ideas closely related to those implied by the theory of supervaluations [4].

To begin, I discuss some issues in the theory of projection that I think require attention. I then alternate between more and less formal discussions of my framework, discussing how it provides the tools to deal with these issues.

## 1.1 Constraining Possible Projection Patterns

The acceptance of the dynamic approach owes a substantial debt to the strong intuitive appeal of Stalnaker’s [13] account of conjunction. Readers have a strong assumption that, given a notion of context change, this account of conjunction is the obvious one. In the context of the pragmatic theory in [13], this intuition may be good enough, but once we situate things in a richer semantic theory like that of [5], and once we include a full range of connectives, quantifiers, and other semantic devices, the obviousness of this particular choice is not reflected in the theory. In particular, for any static, bivalent connective or quantifier, there are a large number of functions potentials compatible with its static bivalent meaning and its type in dynamic semantics, but most possible context change potentials for a given function are never observed. A common hypothetical example (see, e.g. [12]) is the hypothetical connective of *and\**, which differs from the *and* of [5] only in that it updates the context with the second conjunct first and the first conjunct second, but we might also consider more elaborate alternatives, such as an *and<sup>†</sup>* that runs the standard *and* of [5] on the conjuncts and the context, and returns the resulting context unless that is a failure value, in which case it returns the original context unchanged - both of these alternatives agree with the standard *and* so far as truth conditions are concerned, and are readily implemented with dynamic machinery, but are associated with bizarre and apparently unattested projection facts. This problem of over-generation is discussed in more detail by [6] and [11]. A good theory of projection should find a way to constrain the possible connective meanings to reflect the fact that many possible combinations of projection behaviors and truth conditions are not observed.

Stalnaker’s intuition is appealing because it gives us the impression that the projection facts are falling out from the meaning we already knew the connectives had. Once we have our idea of what the mechanisms behind

projection are, and once we know what the connectives mean in static form, we get the projection facts ‘for free’. Predicting projection behavior from basic truth-conditional facts and a general evaluative recipe, in keeping with the intuition of [13], is not a requirement for an acceptable theory, but it is highly desirable. It would be even better if the recipe could be independently motivated. The present discussion does not consider issues of independent motivation, but the framework developed does have the virtue of providing a rule of evaluation from which projection facts can be predicted.

## 1.2 Differences Between Quantifiers

Finally, both the experimental results of Chemla [2] and detailed reflection on various quantifiers reveal that presuppositions in the nuclear scopes of different quantifiers give rise to different presuppositions for the quantified statements and that, in particular, presuppositions in a quantifiers nuclear scope do not always give rise to the universal inferences that much of the literature assumes are there. In particular, although [2] reports that speakers fairly reliably infer the universal (1-f) from (1-a) and (1-b), they report such an inference from (1-c), (1-d), and (1-e) only about half the time, even when told to use a fairly inclusive notion of what counts as an acceptable inference:

- (1) a. Each of the ten students has stopped soking.
- b. None of the ten students has stopped smoking.
- c. More than three of the ten students have stopped smoking.
- d. Less than three of the ten students have stopped smoking.
- e. Exactly three of the ten students have stopped smoking.
- f. Each of the ten students has at some time smoked.

This pattern calls for a theory that its capable of making fine-grained distinctions between different quantifiers in terms of their behavior with respect to nuclear scope presuppositions, something that neither dynamic semantics nor transparency theory is equipped to do.<sup>1</sup> The framework that follows, even if it has trouble with some of the distinctions, at least demonstrates that they can be made within a principled theory of projection.

---

<sup>1</sup>Transparency theory produces non-universal presuppositions for some numerical quantifiers, but only to the extent that if the restrictor is too small the sentence may be true or false trivially, and in the examples above the size of the restrictor is controlled to prevent this effect from coming into play.

## 2 Strong Kleene Semantics

### 2.1 The Idea

The idea of the Strong Kleene [8] trivalent semantics is that, to the greatest extent possible in a straightforward compositional theory, we ignore those cases of presupposition failure that do not bear on the final outcome. Thus, when computing the value of a term  $f(\vec{x})$  if there is some item  $\alpha$  in the list  $\vec{x}$  that involves failure and we would need to know the value of  $\alpha$  precisely to be certain about the value of  $f(\vec{x})$ , then  $f(\vec{x}) = \#$ , where  $\#$  is the symbol for presupposition failure. Otherwise,  $f(\vec{x})$  is whatever value we could recover without knowing about  $\alpha$ .

We formalize this by treating presupposition failure as *uncertainty*. In particular, whenever in a meaning we encounter the truth value  $\#$ , we treat it as a value that must be either 0 or 1, but which is, unfortunately, unknown. This failure ‘doesn’t matter’ iff selecting either of the alternative values for the argument would cause the function  $f$  to give us the same results in light of the other arguments. The key notion here is that of the alternative possibilities we consider for a presuppositional item. For truth values, as noted, the alternatives for  $\#$  are 0 and 1. For other types, we have more freedom, but one obvious generalization to functional types ‘ending in  $\tau$ ’ is that we consider all functions that agree with the presuppositional function where it has a non-presuppositional output, and, for every input that it maps to  $\#$ , have an output of either 0 or 1 instead.

This idea is already quite close to the intuition behind the projection claims usually made for exclusive *or*. In particular (2-a) is taken to presuppose (2-b), which already involves a kind of reasoning by alternatives, and which, notably, says that the presuppositions of the second argument to the disjunction are taken into account only when the first argument is false - i.e., when the values of the other arguments tell us they ‘matter’ for the output of the function:

- (2) a. Somaliland is not a monarchy, or the king of Somaliland is incompetent.
- b. Somaliland is not a monarchy or Somaliland is a monarchy and has a king.

Note that the basic strong Kleene semantics is symmetrical - we ask about each argument in light of all the others - so in a function that is symmetrical between its arguments the order will not matter. This may be plausible for *or* - the reversal of the above appears to yield the same presuppositions:

- (3) a. The king of Somaliland is incompetent, or Somaliland is not a monarchy.  
 b. Somaliland is not a a monarchy or Somaliland is a monarchy and has a king.

However, other symmetrical functions, like the connective *and*, and the quantifiers *no*, *some*, and *at least one*, are usually judged to produce asymmetrical results. We will come back to the mechanisms for dealing with this once we have an implementation of incrementality. This will require a more precise formalization - a task to which we turn next.

## 2.2 Formal Digression - Types and Denotations

### 2.2.1 Sequences

Sequences are enclosed in curved braces, thus  $(x_1, \dots, x_n)$  is a sequence of  $n$  objects. Sequences may also be denoted with an arrow over a letter:  $\vec{x}$  is presumed to be the name of a sequence and the  $i^{\text{th}}$  element of  $\vec{x}$  is presumptively called  $x_i$  for some  $i$  between 1 and  $|\vec{x}|$  (the length of  $\vec{x}$ ). Sequence concatenation will be written with concatenation of names when no confusion is likely, thus, where  $|\vec{x}| = m$  and  $|\vec{y}| = n$   $\vec{x}\vec{y} = (x_1, \dots, x_m, y_1, \dots, y_n)$ . Singleton sequences will frequently and shamelessly be conflated with their sole elements.

### 2.2.2 Types

A *type basis*  $\beta$  is a list of unanalyzed strings. The elements of  $\beta$ , in particular, do not as strings contain any kind of comma, bracket, or arrow symbols. A type basis is called *standard* if it contains at least the symbol  $\mathfrak{t}$ .

For  $\beta$  a type basis,  $T_\beta$ , the *type system generated by*  $\beta$ , is the minimal set satisfying the following closure properties:

1.  $\beta \subseteq T_\beta$ .
2. For all  $n \geq 1$ , all  $\vec{\tau} \in T_\beta^n$ , all  $\tau' \in \beta$ ,  $\langle \vec{\tau} \rightarrow \tau' \rangle \in T_\beta$ .<sup>2</sup>

In the present discussion, we will only consider types ending in  $\mathfrak{t}$  - such as the types associated with common nouns, verb phrases, generalized quantifiers, determiners, and truth-functional connectives.

---

<sup>2</sup>This is an  $n$ -ary function type. It will be convenient, and for the moment not too detrimental, to assume that all function types have basic types as outputs, and that arguments have a canonical order.

### 2.2.3 Denotations

For  $\beta$  a type basis, a *semantic basis for  $\beta$*  is a tuple<sup>3</sup>  $\mathfrak{B} = (\{D_\tau \mid \tau \in \beta\}, \{\#_\tau \mid \tau \in \beta\})$  such that:

1. The  $D_\tau$  for all  $\tau \in \beta$  are all nonempty sets none of whom have any elements that are functions.<sup>4</sup>
2. The  $\#_\tau$  for all  $\tau \in \beta$  are distinct objects, none of which are functions, and none of which belong to  $D'_\tau$  for any  $\tau' \in \beta$ .<sup>5</sup>

We will assume throughout this paper that  $D_\tau = \{0, 1\}$ .

Given a denotation basis  $\mathfrak{B}$  for  $\beta$ , every type  $\tau \in T_\beta$  or list of types  $\vec{\sigma} \in T_\beta^n$  (for  $n \geq 2$ ) is associated with a set of possible denotations  $\Delta_\tau$  or  $\Delta_{\vec{\sigma}}$  and a set of possible *proper* denotations  $D_\tau$  or  $D_{\vec{\sigma}}$ .<sup>67</sup> Beginning with the proper denotations, we have:

1. For each  $\tau \in \beta$ ,  $D_\tau$  is as given in  $\mathfrak{B}$ .
2. For  $n \geq 1$ ,  $\vec{\sigma} \in T_\beta^n$ ,  $D_{\vec{\sigma}} = D_{\sigma_1} \times \dots \times D_{\sigma_n}$ .
3. For  $n \geq 1$ ,  $\vec{\sigma} \in T_\beta^n$ ,  $\tau \in \beta$ ,  $D_{\langle \vec{\sigma} \rightarrow \tau \rangle} = D_\tau^{D_{\vec{\sigma}}}$ .

From these we define the sets of all denotations:

1. For each  $\tau \in \beta$ ,  $\Delta_\tau = D_\tau \cup \{\#_\tau\}$ .
2.  $n \geq 1$ ,  $\vec{\sigma} \in T_\beta^n$ ,  $\Delta_{\vec{\sigma}} = \Delta_{\sigma_1} \times \dots \times \Delta_{\sigma_n}$ .
3. For  $n \geq 1$ ,  $\vec{\sigma} \in T_\beta^n$ ,  $\tau \in \beta$ ,  $\Delta_{\langle \vec{\sigma} \rightarrow \tau \rangle} = \Delta_\tau^{D_{\vec{\sigma}}}$ .

Note the mixing of  $D$ s and  $\Delta$ s in the last clause of the above - it is an important restriction of the present approach that there may be types for presupposition *triggers* but no type allows denotations that act as presupposition *filters*.

<sup>3</sup>The notations  $\{D_\tau \mid \tau \in \beta\}$  and  $\{\#_\tau \mid \tau \in \beta\}$  are shorthand to indicate that the sets in question are sets indexed by the elements of  $\beta$ . More formally, we have two sets  $X_D$  and  $X_\#$  and two functions  $f_D$  and  $f_\#$  from  $\beta$  onto those sets, and, for every  $\tau \in \beta$ , we define  $D_\tau = f_D(\tau)$  and  $\#_\tau = f_\#(\tau)$ .

<sup>4</sup>The  $D_\tau$ s are the denotation domains for the base types.

<sup>5</sup>The  $\#_\tau$ s are the ‘error values’ representing presupposition failure.

<sup>6</sup>The proper denotations are, intuitively, the denotations that are free of presuppositions. If we understand error codes as an encoding of partiality, they are the total functions.

<sup>7</sup>The convention of using corresponding Greek and Latin capital letters for corresponding sets of denotations and partial denotations will be used whenever it seems expedient to define such corresponding sets.

### 2.2.4 Extensions of a Denotation

Since the strong Kleene semantics works by reasoning about sets of alternatives, we will need to define the set of possible alternatives associated with each denotation. Understanding  $\#$  as a ‘gap’ value, and failure as a kind of partiality, we think of this as a process of *extending* the denotation. Thus, where  $\alpha \in \Delta_\tau$  or  $\vec{\alpha} \in \Delta_{\vec{\tau}}$ , we define  $E(\alpha)$ , the set of *proper extensions* of  $\alpha$ , as follows:

- For  $\tau$  a basic, if  $\alpha = \#$  then  $E(\alpha) = D_\tau$ , and if  $\alpha \neq \#$  then  $E(\alpha) = \{\alpha\}$ . (That is, an extension of an object of a basic type that suffers from presupposition failure could be *any* possible denotation for that type, but the only extension of a proper )
- For  $\tau = \langle \vec{\sigma} \rightarrow \sigma' \rangle$ ,  $E(\alpha)$  is the set of all  $\alpha' \in D_\tau$  such that, for all  $\vec{x} \in D_\sigma$ , either  $\alpha'(\vec{x}) \in E(\alpha(\vec{x}))$ . (This is the natural way to type raise the notion of a proper extension of a basic type to a proper extension of an arbitrary type.)
- For  $E(\vec{\alpha}) \in \Delta_{\vec{\tau}}$ ,  $E(\vec{\alpha}) = (E(\alpha_1), \dots, E(\alpha_{|\vec{\alpha}|}))$ . (That is, an extension of a list is a list of extensions).

Having defined our basic formal machinery, we can now give a precise definition of strong Kleene evaluation.

## 2.3 Formal Definition of Strong Kleene Evaluation

For for  $f$  a function and  $\vec{w}$  a list of proper denotations of suitable types,  $f(\vec{w})$  is classical function definition. We define  $f(\vec{u})$  for arbitrary denotations (that is, for all  $\vec{u} \in \Delta_{\vec{\sigma}}$ , where  $f \in \Delta_{\langle \vec{\sigma} \rightarrow \tau \rangle}$ ) in terms of the case of proper denotations as follows:

- If there is  $\alpha \in \Delta_\tau$  such that, for all  $\vec{v} \in E(\vec{u})$ ,  $f(\vec{v}) = \alpha$  (for the usual definition of  $f(\vec{v})$ ), then  $f(\vec{u}) = \alpha$ .
- Othewise,  $f(\vec{u}) = \#$ .

## 2.4 Predictions of the Basic Strong Kleene System

### 2.4.1 Calculating Denotations and Presuppositions

In practice, when we calculate the presuppositions of a sentence, we ask two questions: ‘what is necessary for the sentence to be true?’ and ‘what is

necessary for the sentence to be false?’ These are fairly easy to determine under the system we’re discussing, since the truth-conditional semantics of the functions involved and a little imagination about possible extensions is all that is required. Once we have answers to both of these questions, we know the sentence presupposes the disjunction of the answers, which describes exactly those cases in which the sentence fails to take the truth value #.

### 2.4.2 Connectives

First, consider the sentence (4)

(4) John is incompetent and he knows it.

The first clause of (4) takes the value 1 if John is incompetent and the value 0 otherwise. The second clause takes the value # if John is not incompetent, the value 1 if John is incompetent and is aware of his own incompetence, and the value 0 if John is incompetent but unaware of his own incompetence. We now ask two questions:

when is (4) true? Where our function  $f$  is truth-functional conjunction and  $x$  and  $y$  are proper denotations,  $f(x, y) = 1$  if  $x = y = 1$ . So, for (4) to take the value 1, we must require both that the first argument has the value 1 and that all extensions of the second argument have the value 1. That is, both arguments must be 1 (in particular, in the case where the second argument is #, we have  $E(\#) = \{0, 1\}$ , but  $f(1, 0) = 0 \neq 1$ ). So, for truth we require that the first conjunct be true (John is incompetent) and that the second conjunct be true as well (John is incompetent and aware of his incompetence).

when is (4) false? Whenever either conjunct has the value 0, the sentence has the value false, since, no matter what options there are for the other argument, anything combined with 0 by logical conjunction gives us 0. Since we’ve already addressed the case where both arguments are 1, and the first argument is non-presuppositional, the only remaining case is the case where the first argument is true and the second suffers presupposition failure, but here 1 is among the possible extensions of # and  $f(1, 1) \neq 0$ , so in this case the sentence cannot be false. Thus, the sentence is false whenever John is not incompetent or John is incompetent but doesn’t know it.

what does (4) presuppose? Sentence (4) presupposes that either John is competent or John is

incompetent but doesn't know it or John is incompetent and does know it. These conditions together cover all alternatives, so the sentence is effectively presuppositionless. This is consistent with the usual view expressed in the literature. In general, sentences of the same form as (4), but with a second conjunct with presuppositions less trivially connected to the truth conditions of the first conjunct, will presuppose that either the first conjunct is false or both the first conjunct and the presuppositions of the second conjunct are true. But this just amounts to saying that the relation of the material conditional holds between the first conjunct and the presuppositions of the second.

Unfortunately, since the theory is symmetrical, it makes the same prediction of presuppositionlessness if we reverse the order of the conjuncts in (4):

- (5) John knows that he is incompetent, and he is.

However, (5) is widely regarded as presupposing that John is incompetent (or at least as being quite odd), so the symmetry of the simple strong Kleene theory appears to be a liability.

Of course, symmetry might be desirable in some other cases, as with (6):

- (6) a. The bathroom is well hidden, or there isn't one.  
 b. There isn't a bathroom, or the bathroom is well hidden.

The strong Kleene procedure predicts that both examples in (6) will have no presuppositions, since, where  $f$  is logical disjunction  $f(1, 0) = f(1, 1) = 1$ , so  $f(1, \#) = 1$ , and likewise for  $f(\#, 1)$ , but  $f(0, \#) = \#$  since  $f(0, 1) \neq f(0, 0)$ . Thus, if there is no bathroom, then in both of the above one if the disjuncts is true so the sentence is true, and if there is no bathroom, then the disjunct about there being no bathroom is false, but the presuppositions of the other disjunct are met, so the sentence will be either true or false depending on how well-hidden the bathroom is. For a sentence of the form  $p$  or  $q$  where  $q$  presupposes that  $q'$  and  $p$  has no presuppositions, we in general get a presupposition that if  $p$  is false then  $q'$  is satisfied, as with the projection behavior attributed to *or* in [1] and [10], unlike these works, we also get the flipped version, which may be more desirable in the case of examples like (6) then it was for cases of conjunction.

For *if*, analyzed as the material conditional, our predictions are similar to those for *or*. In a sentence of the form *if*  $p$ ,  $q$ , the sentence is true so long as  $p$  is false or  $q$  is true, so if either of these is the case, any presupposition failure in the other argument will be irrelevant. However, if  $p$  is true or  $q$

false, then the presuppositions of the other must be met, because changing the truth value of the other argument in this case will affect the outcome. In particular, we predict that *if #*, 1 is true (which is not in keeping with standard assumptions, but is hard to conclusively rule out), and that *if 0*, # is true as well. For a sentence like (7), we find a presupposition that if France is a monarchy then it has a king (because if France is not a monarchy the sentence is trivially true):

(7) If France is a monarchy then the king of France is bald.

So, again, the predictions of the strong Kleene theory align well with the received wisdom for presuppositions of the second argument of a connective, but less well for presuppositions of the first argument.

### 2.4.3 Quantifiers

For the nuclear scopes of quantifiers, the predictions of the strong Kleene theory are, with a few exceptions, plausible. Let's consider the predictions for various classes of quantifiers with a series of explicit examples.

First, consider the quantifier *each* as in (8):

(8) Each of the ten students has stopped smoking.

when is (8) true? Let  $q$  be the denotation of *has stopped smoking*, and  $p$  be the characteristic function of the set of the ten students. For all  $q' \in E(q)$ , it must be that  $all(p, q') = 1$ . Now,  $all(p, q') = 1$  iff  $q'(x) = 1$  for every  $x$  such that  $p(x) = 1$ , and this is true for all  $q'$  iff it is true of  $q$  (since if there is  $x$  such that  $p(x) = 1$  and  $q(x) = \#$ , then there is  $q' \in E(q)$  such that  $q'(x) = 0$ , and if there is  $x$  such that  $p(x) = 1$  and  $q(x) = 0$ , then for all  $q' \in E(q)$ ,  $q'(x) = 0$ ). Hence, for (8) to be true, it must be true of each of the ten students that he or she once smoked but no longer does, so in particular for (8) to be true, the universal claim that each student once smoked must also be true.

when is (8) false? With  $p$  and  $q$  as above, we note that, for any  $q' \in E(q)$ ,  $all(p, q') = 0$  iff there is some  $x$  such that  $p(x) = 1$  (i.e.  $x$  is one of the ten students) but  $q'(x) = 0$ . So to make (8) false, we demand this of every  $q' \in E(q)$ , and for every  $q'$  to have this property, we need only for there to be one student that  $q$  maps to 0, so (8) is false whenever there is one student who has smoked before and still does.

Given the above conditions, (8) presupposed that either every student is a former smoker or one student is a current smoker who has smoked

in the past. Thus, we get a disjunction of two conditions, one existential and associated with falsehood, and the other universal and associated with truth. This prediction is consistent with the data on universal presuppositional inferences from positive assertions, but is, perhaps, too weak overall. Later on, we shall see one way to modify the theory to strengthen these presuppositions.

Repeating the exercise with the nuclear scope of *none* in (9), we get similar results:

(9) None of the ten students has stopped smoking.

when is (9) true? For *nonesmoking* to be true, given  $p$ , and  $q$  as above, we need it to be the case that for all  $q' \in E(q)$ , there is no  $x$  such that  $p(x) = 1$  and  $q'(x) = 1$ . That is, for all  $q' \in E(q)$  and all  $x$  such that  $p(x) = 1$ ,  $q'(x) = 0$ . This is true iff  $q(x) = 0$  for all  $x$  among the ten students, since if there is any student that gets mapped to 1, things obviously don't work, and if any student gets mapped to  $\#$ , there will be some extension of  $q$  that maps that student to 1. This means that, for the sentence to be true, nuclear scope must be false of all the students, and so its presupposition must be true of all the students - again, we get a universal inference.

when is (9) false? For *nonesmoking* to be false, what we need is for there to be one student who is a former smoker. So long as there is one such student, every extension of the nuclear scope's denotation will make the *none* quantification false, but, if there is no such student, then there will be at least one extension that maps every student to 0, making the *none* quantification true.

We find, in short, that (9) presupposes that either all students are former smokers or at least one student is a one-time smoker who still smokes. Again, this is weaker than the commonly assumed universal presupposition, but still strong enough that the universal statement follows from the sentence being true.

Before turning to the numerical quantifiers studied by [2], let's briefly consider existential quantification with *at least one*:

(10) At least one of the ten students has stopped smoking.

when is (10) true? To make (10) true, all that is needed is a single former smoker among the ten students, since any function for the nuclear scope that sends one of the students to 1 will suffice to make the sentence true.

when is (10) false? For the same reason, to make (10) false, every student must be a current smoker with some smoking history. If one student has never smoked or is a former smoker, then there are some extensions of the denotation of the nuclear scope that make the sentence true.

The above tells us that (10) presupposes that either one of the students is a former smoker or all of them have been and remain smokers - thus we get only a very weak existential inference from the truth of the sentence. It should be noted that, for cases where the size of the restrictor is not specified explicitly, the emptiness of the restrictor is in fact sufficient to make existential quantification false.

Now, we turn to more complex cases of numerical quantification, where as before  $p$  is the restrictor that picks out the ten students under discussion, and  $q$  is the trivalent predicate associated with the expression *have/has stopped smoking*. First, consider *less than*:

(11) Less than three of the ten students have stopped smoking.

when is (11) true? (11) is true whenever no  $q' \in E(q)$  maps three or more of the ten students to 1. To achieve this, we need to map at least 8 of the ten students to 0. If 7 or fewer students are mapped to 0, then at least three students are mapped to 1 or #, giving us at least one  $q'$  where three or more students are mapped to 1. Thus, (11) is true iff at least eight of the ten students once smoked and still do.

when is (11) false? For the same reason, (11) is false iff three or more of the students are former smokers.

The prediction is that (11) does not support any universal presuppositional inference, although, to give it any chance of being true, we need a majority of the ten students to satisfy the presupposition. Note that, under the system sketched here, a *less than  $n$*  quantification is also true if the cardinality of the restrictor is less than  $n$  - a possibility that does not come up with these examples because the cardinality of the restrictor is fixed with a partitive construction.

Next, consider *more than*:

(12) More than three of the ten students have stopped smoking.

For reasons similar to those given for (11) above, (12) is true iff at least four of the ten students are former smokers, and false of at least seven student who once smoked and still do. In general, a *more than  $n$*  sentence will

be false when the restrictor contains  $n$  or fewer elements, regardless of the presuppositions of the nuclear scope.

Finally, we turn to *exactly*:

(13) Exactly three of the ten students have stopped smoking.

when is (13) true? (13) is true iff for all  $q' \in E(q)$ , exactly three of the ten students are mapped to 1 by  $q'$ . This in turn is the case iff every student has smoked in the past and exactly three of them have stopped. In any other configuration, either the quantification will be false in all variants, or by switching the value assigned to some of the students who have never smoked, we can switch between making the quantification true and false, leading to failure.

when is (13) false? (13) is false iff either more than three of the students are former smokers, or more than seven once smoked and still do, since in either case all alternations will preserve the fact that there are either too many former smokers to make the 3 exact, or too many still-smokers to get us 3 former smokers.

This is problematic, because it predicts that (13) entails that each of the students used to smoke, which runs contrary to intuition and experimental results.

Finally, we should say a word about presuppositions in restrictors. Most of the quantifiers considered above are symmetrical, so that we get analogous presuppositions when we reverse the two arguments. The resulting predictions are rather odd. For the asymmetrical quantifier *each*, consider the following sentence as an example:

(14) Each student who has stopped smoking is participating in the number theory seminar.

The strong Kleene theory predicts that this sentence is true iff every student who is a former smoker or has never smoked is participating in the number theory seminar, and false iff at least one former smoker is not participating in the number theory seminar. Otherwise, it suffers presupposition failure. The predicted presupposition does not, so far as I know, correspond to the main natural intuition.

### 3 Adding Incremental Failure Conditions

We noticed two (related) kinds of problems with the basic strong Kleene system: first, it often predicts presuppositions that are weaker than the ones we would expect, and second, it has no facilities for taking order of arguments into account. In this section, incremental failure conditions are developed as a response to these difficulties.

#### 3.1 The General Idea

An incremental failure condition is a condition on some property of an argument and its presuppositions with respect to the function and the preceding arguments, but independent of the values of the arguments that follow. If any argument in a sequence fails the failure condition, the value of the function at the argument sequence in question will be #, even if the basic strong Kleene system could assign it a proper truth value. Failure conditions straightforwardly make presuppositions stronger by introducing new sources of presupposition failure that must be avoided. Since they have access to information about prior but not subsequent arguments (although, in the formalism developed below, only a fairly limited kind of information), they also introduce asymmetry - even if the basic function is symmetric, the failure condition may know something when it hits a later argument that it didn't know for an earlier one, so order matters.

#### 3.2 Two Kinds of Conditions - Informally

Two kinds of failure conditions that seem promising are presupposition-safety conditions and truth-friendliness conditions. These conditions have intuitive interpretations, although they don't really have strong theory-independent intuitive motivations.

First, a presupposition-safety condition demands that, rather than knowing that an argument's presuppositions don't matter at the end, we must also know that they're guaranteed not to matter at the time we read it in. Thus '*# and 0*' would be assigned the truth value # on such a condition, for, although this expression is assigned a non-# truth value by the basic system, when we read in the first argument, we don't know what the second argument will be. We must consider the scenario in which the second argument turns out to be 1, and, in that scenario, the presuppositions of # spell disaster, since the strong Kleene system is incapable of assigning a non-# truth value to '*# and 1*'. On the other hand with '*0 and #*', we already

know, at the time we encounter  $\#$ , that the other argument is not 0, so we don't need to consider the problematic condition, and we compute a truth value of 0. In this way, this condition restores the generally accepted asymmetric behavior of *and*. This condition is also fairly intuitive in its intent - it takes the intuition of the strong Kleene system - that we only have failure in the function's outputs when we fail to convince ourselves that failure in the arguments doesn't matter - but it says we can't withhold judgment forever - when we encounter a new argument, we must decide right away whether we can convince ourselves that it's 'safe'. This condition agrees precisely with the received wisdom of [5] and [1] for the standard truth-functional connectives - it also reconstructs precisely the trivalent connectives stipulated in [10]. It is closely related to the intuitions behind the 'short circuit logic' of boolean operations in many programming languages - an idea previously investigated in the context of presuppositions in [9].

Another class of conditions can be described as 'output bias' conditions. These demand that some output must never be ruled out by presuppositions alone. The intuition is that, when it encounters a new argument, the semantic evaluator first looks at just the presuppositions of that argument, if these presuppositions make it impossible to achieve some desired output that was left attainable by the function and the previous arguments, failure results. We can imagine the compositional semantics clinging to hope for its preferred output, and being disappointed but accepting when that hope is destroyed by the assertions of an argument, but inconsolable with it is destroyed by something as sneaky as presuppositions.<sup>8</sup> The particular output bias I wish to consider is truth bias - that is, a preference for an output of 1. Of course, since output bias conditions depend on the notion of what possibilities remain open, they are also relative to a lower layer of assumptions about how the semantics works. Two obvious assumptions to give to a truth bias condition are that the semantics is the basic strong Kleene semantics, and that the semantics is the strong Kleene semantics augmented with the presupposition-safety condition above.

---

<sup>8</sup>Compositional output bias, which is investigated here, is not as easy to justify from functional intuitions as a non-compositional version of output bias might be. The exploration of a non-compositional version of the present theory is beyond the scope of this paper, but, at present, I am not aware of any data that favor a non-compositional version, and the compositional treatment is more manageable. However, further investigation of non-compositional filters is needed.

### 3.3 Formal Digression: Incrementality and Failure Conditions

#### 3.3.1 Incremental Implementation of Strong Kleene

To build a system for working with failure conditions, we first incrementalize the strong Kleene system by defining the strong Kleene update of a function (of any number of arguments) by a single (first) argument. This does some of the work of Currying in most semantic theories.

For  $f \in \Delta_{\langle(\sigma_1, \dots, \sigma_n) \rightarrow \tau\rangle}$ , and  $x \in \Delta_{\sigma_n}$ ,  $f(x)$  is the unique function  $g \in \Delta_{\langle(\sigma_2, \dots, \sigma_n) \rightarrow \tau\rangle}$  such that, for all  $\vec{y} \in D_{\langle(\sigma_2, \dots, \sigma_n)\rangle}$ ,  $g(\vec{y}) = f(x, \vec{y})$ , where  $f(x, \vec{y})$  is the strong Kleene application of  $f$  to the sequence  $f(x, \vec{y})$  as defined above. Note that  $g$  is not necessarily a proper denotation, even if  $f$  is. In particular, if  $x$  is presuppositional, there are likely to be some sequences  $\vec{y}$  of proper denotations such that  $f(x, \vec{y}) = \#$  under the the strong Kleene system. Applying  $f$  to its arguments one at a time by this new definition should have the same outcome as applying it to them all together by the old definition.

#### 3.3.2 Working with Failure Conditions

The above definition allows us, proceeding through the arguments, to remember everything we need to know about the impact of each argument on the strong Kleene evaluation of the function, but nothing else. If more than one combination of past arguments will have the same interaction with any future arguments under the strong Kleene system, we don't remember which one it was. This provides a constraint on the system. Failure conditions are invoked when we apply a function to a single argument, so we only know about the content of the previous argument to the extent that it has shaped the function we have before us. Formally, a failure condition is a class of pairs of the form  $\langle f, x \rangle$ , where  $f$  is a function and  $x$  is an improper denotation for the first argument type of  $f$ , encoding the 'is a good argument for' relation between denotations and functions - a function fails when it is given a bad argument - that is, one not paired with it by the failure condition. We might impose many restrictions on acceptable failure conditions (which as defined above are too powerful), but in particular we should insist that proper denotations never trigger a failure condition. Even with these constraints, failure conditions are an incredibly powerful parameter that lets us generate a wide variety of theories - it is, obviously, desirable to pick relatively succinct failure conditions.

For  $\vec{\eta}$  a list of failure conditions, an  $x$  a denotation of the first argument

type of  $f$ ,  $f_{\vec{\eta}}[x]$  - the *deployment of  $f$  on  $x$  under failure conditions  $\vec{\eta}$*  - is defined as follows:

- If  $x$  and  $f$  fail some failure condition  $\eta_j$  in the list  $\vec{\eta}$ ,  $f_{\vec{\eta}}[x] = \#$ .<sup>9</sup>
- Otherwise,  $f_{\vec{\eta}}[x] = f(x)$ .

Within this framework, a projection theory is a selection of the failure conditions that are at work in the language.

Consider now, as an example, the case where  $\eta$  consists of just the presupposition-safety condition sketched informally above. Let  $f$  equal normal boolean *and*, and consider  $f_{\eta}[\#][0]$  (recalling that  $f_{\eta}(\#)(0) = 0$  under the basic strong Kleene system). We can see that  $f$  and  $\#$  fail to meet the presupposition-safety condition, since there is some possible further argument, in particular 1, for which  $f(0)(1) \neq f(1)(1)$ , where 0 and 1 are both possible proper extensions of  $\#$ , so, at the point in the derivation where we encounter the first  $\#$ , its presuppositions could turn out to matter, so it fails the filter and we evaluate  $f_{\eta}[\#]$  to  $\#$ .<sup>10</sup>

### 3.3.3 Presupposition-Equivalence

Before defining some failure conditions we might want to use, we will need one additional piece of equipment. Recall that the output bias conditions require us to reason about what a function makes possible or impossible by virtue of its presupposition alone. To clarify the notion of ‘by virtue of its presuppositions alone’, it will be useful to be able to talk about when two denotations of the same type are presupposition-equivalent. For  $x, y \in \Delta_{\tau}$  for some type  $\tau$ ,  $x$  is presupposition-equivalent to  $y$ , written  $x \simeq y$ , iff either  $\tau$  is a basic type and  $x = \#$  iff  $y = \#$ , or  $\tau$  is a function type  $\langle \vec{\sigma} \rightarrow \sigma' \rangle$  and for all  $\vec{w} \in D_{\vec{\sigma}}$ ,  $x(\vec{w}) = \#$  iff  $y(\vec{w}) = \#$ . This is just the straightforward definition that says two things are presupposition equivalent if they fail under the same circumstances.

### 3.3.4 Some Failure Conditions Defined

Finally, we can precisely define the failure conditions sketched above. First, a function  $f \in \Delta_{\langle \vec{\sigma} \rightarrow \tau \rangle}$  and an argument  $x \in \Delta_{\vec{\sigma}_1}$  are *presupposition safe* iff for

---

<sup>9</sup>Here  $\#$  is used as a shorthand for the truth value  $\#$  or for any function of appropriate type whose output is always  $\#$ .

<sup>10</sup>That is, the constant function of type  $\langle \mathfrak{t} \rightarrow \mathfrak{t} \rangle$  which always outputs  $\#$ .

all  $\vec{y} \in D_{(\sigma_2, \dots, \sigma_{|\vec{\sigma}|})}$ , and all  $x', x'' \in E(X)$ ,  $f(x', \vec{y}) = f(x'', \vec{y})$ . The presupposition safety condition says that a pair fails if it is not presupposition-safe. We call this condition  $\gamma$

Second, a function  $f$  and a function  $f \in \Delta_{(\vec{\sigma} \rightarrow \tau)}$  and an argument  $x \in \Delta_{\sigma_1}$  are *weakly truth-friendly* iff either there are  $x' \simeq x$  and  $\vec{y} \in D_{(\sigma_2, \dots, \sigma_{|\vec{\sigma}|})}$  such that  $f(x', \vec{y}) = 1$  or there is no  $\vec{w} \in D_{\vec{\sigma}}$  such that  $f(\vec{w}) = 1$ . This requirement says that there must be some way we could have an argument with the presuppositions of  $x$  and still get an output of 1 out of the strong Kleene system, unless of course we knew we couldn't get an output of 1 before we even got to  $x$ , in which case  $x$  and its presuppositions are hardly blameworthy. As a failure condition, we say that a function and argument pair fails if it fails to be weakly truth-friendly. We call this condition  $\delta$ . It might at first appear that applying this condition prevents us from ever computing an outcome other than  $\#$  and 1, but this is not the case - we can still get an output of 0, from the step where we compute  $f(x)$ , when the assertive component of  $x$  comes into play.

Finally we have the *strong truth-friendliness* condition, which is the result of modifying the weak truth-friendliness condition to use evaluation with presupposition-safety, rather than ordinary strong Kleene evaluation, in its determination of what is possible. A function  $f$  and a function  $f \in \Delta_{(\vec{\sigma} \rightarrow \tau)}$  and an argument  $x \in \Delta_{\sigma_1}$  are *weakly truth-friendly* iff they are presupposition-safe and either there is no  $\vec{w} \in D_{\vec{\sigma}}$  such that  $f(\vec{w}) = 1$ , or else there there are  $x' \simeq x$  and  $\vec{y} \in D_{(\sigma_2, \dots, \sigma_{|\vec{\sigma}|})}$  such that  $f$  and  $x'$  are presupposition-safe and  $f(x', \vec{y}) = 1$ . This is a condition I've used in the past, so I've included it in the interest of completeness, but I don't plan to discuss it more here.

### 3.4 Predictions from the Presupposition-Safety Condition

The presupposition-safety filter introduces asymmetry for commonly studied propositional connectives. In the cases of *and*, *or*, and *if*, presupposition failure in the first argument will not be tolerated, since for all these connectives, there is some potential second argument for which switching the value of the first argument will switch us between truth and falsehood - thus, we predict that presuppositions project unchanged from the first arguments of all these connectives. In the second argument, however, we recover the weaker presuppositions of the basic strong Kleene theory. Thus, in particular, if the first conjunct of an *and* or *if* sentence has value 0, the value of the whole expression is already determined (to be 0 for *and* and 1 for *if*), so the value of the second argument only affects the truth value of the coordinated

expression if the truth value of the first argument is 1 - so for both *and* and *if* we have the conditional presuppositions reported in the literature. For *or* a 1 in the first argument makes the second argument's truth value irrelevant, so we have that a presupposition in the second argument projects only conditioned on the negation of the first argument - again agreeing with the received view of much of the literature.

For the restrictors of the quantifiers *no*, *some*, and *all*, the presupposition-safety condition will cause presupposition failure if the predicate used as the restrictor maps *any* entity to #, since whether we map that entity to 0 or 1 (that is, whether we include it in the restrictor) can be made relevant by selecting a hypothetical second argument (nuclear scope) that has uniform behavior on all other entities and deviates only for the entity in question. For *no* and *some*, we select the predicate that maps only the distinguished entity to 1, and all other entities to 0. For *all*, we reverse this - in all cases, the choice of including or excluding the selected entity when evaluating the restrictor will switch the resulting quantification between two truth values, violating the safety condition. For certain numerical quantifiers, we do not get quite this strong of an effect - in particular if the numerical quantifier is of the form *at least n*, *more than n*, or *exactly n*, then if the total number of individuals that make the restrictor maps to 1 or # is too small, the sentence will simply be false. Likewise, for *less than n* and *at most n*, the sentence may be true so long as the restrictor fails in few enough cases to keep the total number of entities it maps to 1 or # sufficiently small. As mentioned above, many presuppositions in restrictors appear not to result in any strong presuppositions for the resulting quantified sentence, so these predictions are suspect.

Finally, for the presuppositions in the nuclear scopes of quantifiers, the presupposition-safety condition does not affect our predictions - since the nuclear scope is the final argument, a presupposition there will be violated iff it resulted in failure under the regular strong Kleene condition. Those predictions were, for the most part, quite reasonable (except in the case of *exactly*, where they were too strong), although we might have preferred universal presuppositions (instead of just universal inferences) for *each* and *none*.

### 3.5 Predictions from the Weak Truth-Friendliness Condition

Weak truth-friendliness makes some more eccentric predictions. With connectives, it preserves the symmetry of *or* from the basic strong Kleene system, since if # is the first argument of the disjunction there's still a

possible second argument (namely 1) that will make it true. Likewise, it still evaluates *if* #, 1 as true. However, *and* is now asymmetric. *and*[#] will result in failure since *and* can output 1, but *and*(#) cannot, an # is presupposition-equivalent only to itself - thus *and*[#][0] = #. On the other hand, *and*[0][#] = 0, since *and*[0] is a function that cannot output 1, so when we apply it to the argument # truth-friendliness is satisfied trivially by the rule that says the function we had before had to give us some hope of truth. Thus, we now get the asymmetric *and* of [10], [5], and others, for which a presupposition in the first conjunct projects, while a presupposition in the second conjunct projects only conditionally. This prediction is not in keeping with normal claims for *if*, but it does capture an apparently real difference between *and* (which appears asymmetrical in its projection of presuppositions) and *or* (which often shows symmetrical behavior).

Turning to quantifiers, we find that almost any nontrivial value for a restrictor will generally be truth-friendly, since the great freedom to pick a hypothetical second argument (predicate to serve as the nuclear scope) will allow us to map all the entities that the restrictor maps to # to either 0 or 1, one or the other of which will generally permit us to ignore them when evaluating truth. Further, since the condition requires only that we achieve truth with some function that is presupposition-equivalent to the restrictor, we can tailor truth and falsehood as needed. In fact, unless the set of entities is too small for the numerical portion of the quantifier to work out or the restrictor suffers failure for almost all entities, the only quantifiers that will give us trouble for this condition are proportional ones like *exactly half*. Thus, except in most cases, we get the same odd behavior we got out of the basic strong Kleene system.

For nuclear scopes, things get more interesting. Now, with the restrictor already in hand, we ask whether the presuppositional part of the nuclear scope is enough to rule out truth. For *none* and *each*, any presupposition failure of the nuclear scope for any item in the restrictor is enough to rule out truth, no matter what we do with the assertive component, so these quantifiers now produce genuine universal presuppositions, failing whenever the nuclear scope fails on the restrictor. For *exactly three* the story is the same - here too there is no possible nuclear scope that makes the quantifier true and maps any element of the restrictor to #, so if the nuclear scope maps any element of the restrictor to #, truth-friendliness fails (unless the restrictor is so small that falsehood was already guaranteed, of course). For the other quantifiers we've considered, we still fail to achieve universal presuppositions - for all of these, we only require that there be some value of the nuclear scope that would result in truth under the strong Kleene system

and that maps the same items as the actual nuclear scope to  $\#$ . So, for *at least one*, we add the trivial requirement that at least one item in the restrictor must not suffer failure under the nuclear scope, for *less than three of the ten students*, we require that at least eight of the ten students satisfy the presuppositions of the nuclear scope, and for *more than three of the ten students*, we require that at least four of the students satisfy the nuclear scope’s presuppositions. In all cases, these requirements must be satisfied to get either truth or falsehood, in addition to the requirements for truth or falsehood imposed by the strong Kleene theory.

Of the two theories filters explored here, I think the second is more promising, although its behavior with *exactly* and restrictors remains problematic.

## 4 Issues

### 4.1 Order and Syntactic Interface

The notion of order of arguments is left deliberately abstract above. However, it is unlikely that it is pure hierarchical order. In particular, if we accept any of the syntactic theories of conjunction that says *and* forms a constituent with its rightmost argument, and we apply the function to its arguments in the usual hierarchical order (that is, it combines sooner with arguments with which it forms smaller constituents), any of the asymmetric filters discussed above will project all presuppositions of the *second* conjunct, while projecting presuppositions in the first conjunct only conditionally with respect to the second. It might be that the proper syntax avoids this issue, but I suspect it will be better to say that within particular syntactic domains left-to-right order is preferred over hierarchical order. This is not in keeping with most definitions of ‘strong compositionality’, but, so long as there is a bound on the domains within which this left-to-right order is applied, it preserves most of the theoretical, intuitive, and practical advantages of a compositional theory (including the relative tractability of computing denotations and producing inductive proofs about the semantics, and perhaps many of the learnability benefits as well).

### 4.2 Problematic Data

#### 4.2.1 ‘exactly’

The strong Kleene system, with or without additional failure conditions, predicts universal inference from presuppositions in the nuclear scope of an

*exactly* quantifier, so (15-a) yields an inference (though not necessarily a presupposition) of (15-b) in all versions of the theory presented here.

- (15) a. Exactly three of these students have stopped smoking.  
b. Each of these students has stopped smoking.

To see why this is so, we ask when (15-a) is true under the strong Kleene theory. The first argument, the set of the students, has no proper extension but itself (on the assumption that the demonstrative is resolved successfully), but the second argument will have many if some students have never smoked - for the sentence to be true, each of these extensions must make the quantifier give a value of 1. If more than three students are former smokers, then any extension maps more than three students to 1, so all of them make the sentence false. If fewer than three students are former smokers, then at least one extension (the one where we replace every output of # with 0) makes the sentence false, so it is untrue. If exactly three student are former smokers and at least one student has never smoked, then when we map all the students who have never smoked to 0, we make the sentence true, but when we map them to 1, we push the total number of instances of 1 above three, making the sentence false. The only case left is the case where exactly three students are former smokers and all the rest once smoked and still do - in this case, there is only one proper extension and it makes the sentence true. Thus (15-a) is true iff exactly three of the students are former smokers and all the rest have smoked and still smoke. Thus (15-a) entails that all students either once smoked and now don't or once smoked and still do, which is to say it entails (1-a).

Of course, not all of the alternatives make this universal inference a presupposition, but that they make it an entailment is bad enough - I find it perfectly reasonable to utter (1-e) in a world where some of the students under discussion have never smoked, and the data reported in [2] indicate that about half of native French speakers fail to recognize such a presuppositional inference. Indeed, *exactly* sentences are among the sentences that make the most speakers I've talked to willing to accept weak or trivial presuppositions. The fact that it is here associated with the strongest inferences is therefore a serious empirical problem for any kind of strong Kleene theory.

One possible confounding factor with *exactly* is that the analysis of *exactly three* or even *exactly three of these students* as a constituent contributing simple quantificational force is questionable. If *exactly* is recognized as an adverb - perhaps one adjoined to a much larger constituent, or that has some sort of metalinguistic force - then the implications of this theory are

much less clear-cut.<sup>11</sup> Interestingly, *only* - an adverb in certain respects similar to *exactly* appears to give rise to much stronger projections of nuclear scope presuppositions than *exactly*:

- (16) a. Exactly three of these students have stopped smoking.  
b. Only three of these students have stopped smoking.

I think there is a strong intuition that, while (16-a) does not presuppose that all the students have smoked, (16-b) does, or at least imposes some stronger requirement - however, in cases where non-presuppositional arguments are involved, the two are true under quite similar circumstances. This is most likely best attributed to the famously difficult behavior of *only*, which is, among other things, a presupposition trigger in its own right, but it still needs further investigation.

In any case, the current system is incapable of accounting for the *exactly* data without further augmentation. In searching for ways to augment the system, it will, I think, be useful to note that our intuitions that (16-a) is true so long as exactly three students are former smokers, whether the others have ever smoked or not, are predicted if we simply allow *exactly* to treat cases of presupposition failure in the nuclear scope as cases of falsehood.

#### 4.2.2 restrictors

There are many cases where the projection of presuppositions from restrictors seems quite weak. In particular, the following seems to project no substantial presupposition from the restrictor, but only to assert something about the set of countries that have kings and respect them:

- (17) Every country in the world that respects its king has agreed to the treaty.

In particular, for the above to be true it does not need to be the case that the countries that do not have kings have agreed to the treaty.

More strongly, the following are, I think, judged true so long as no student former smoker has taken the phonetics exam:

- (18) a. No student who has stopped smoking has taken the phonetics exam.

---

<sup>11</sup>Giorgio Magri and Philippe Schlenker have both raised concerns along these lines, and to Magri in particular for suggesting the comparison with *only*.

- b. None of these students who has stopped smoking has taken the phonetics exam.

As noted by [11], it is hard to tell whether this is in fact an effect of restrictors or of relative clauses, but it is, in any case, rather pronounced. Here again, even the basic version of the strong Kleene theory seems to produce presuppositions that are too strong. Also, just as in the case of *exactly*, we get the right predictions if we allow ourselves to treat presupposition failure as falsehood for the purposes of evaluating restrictors.

### 4.2.3 possible responses

Both the *exactly* problem and the problem of restrictors involve cases where the actual truth-conditions we observe are about what we would expect if presupposition failure were associated with falsehood. One possible solution is to say that under certain circumstances we can (or perhaps even must) insert into the logical form a ‘flattening’ operator which converts presuppositional functions to bivalent functions by replacing all # output values with 0. The key problem for this approach would be explaining why it is difficult or impossible to insert this operator to block presuppositions in the nuclear scope of *no*.

Doing similar work to the above without an explicit operator, we could say that the set  $E(f)$  need not include all functions that agree with  $f$  for all  $x$  such that  $f(x) \neq 0$ , but may rather include only a handful of them, and, in particular, may in some cases include only one, where in the case of *exactly* and restrictors that  $E(f)$  may be a singleton set containing only the function  $f'$  such that  $f'(x) = 1$  iff  $f(x) = 1$  and  $f'(x) = 0$  iff  $f(x) \neq 1$ . This set would be expanded only under certain circumstances - say, only when  $f'(x) = 0$  for all  $x$  in some set of ‘relevant values’. If the set of relevant values for the nuclear scope of a quantifier is all elements in the restrictor, this line of thinking can help us to distinguish between *exactly* and *no*. I hope to pursue this line of inquiry in future work.

Another approach is to say that both of these classes of expressions are special or metalinguistic, and are permitted to step in some way outside the mechanisms described above and to manipulate the presuppositions more directly. The idea that restrictors are a special phenomenon does have some intuitive appeal, and likewise *exactly* has many adverbial qualities and could be understood as saying of the proposition that it is true with respect to some parameter ‘by the skin of its teeth’.<sup>12</sup> Thus *exactly three* sentences could

---

<sup>12</sup>I am grateful to Philippe Schlenker and Giorgio Magri for helpful discussions of this

be associated with, and get their presuppositions from, the corresponding sentences that use only *three*, but then strengthen them by demanding that 3 be the weakest number that would make the proposition true. Pursuing this line would, of course, require describing of a component of a semantics to handle such operations.

## References

- [1] Beaver, D. 2001. *Presupposition and Assertion in Dynamic Semantics*. CSLI.
- [2] Chemla, E. 2007. ‘Presuppositions of quantified sentences: Experimental Data’, Ms.  
<http://www.emmanuel.chemla.free.fr/Material/Pres-Exp.pdf>
- [3] Fox, D. 2007. Manuscripts on presupposition projection.
- [4] van Fraassen, B. 1966. ‘Singular Terms, Truth Value Gaps, and Free Logic’. *The Journal of philosophy* 63;17.
- [5] Heim, I. 1983. ‘On the Projection Problem for Presuppositions’, WCCFL 2.
- [6] Heim, I. 1990. ‘Presupposition Projection’. In R. van der Sandt (ed.), *Reader for the Nijmegen Workshop on Presupposition, Lexical Meaning, and Discourse Processes*. U. of Nijmegen.
- [7] LaCasse, N. 2007. Notes distributed at UCLA presupposition seminar.
- [8] Kleene, S. 1952. *Introduction to Metamathematics*. North Holland, Amsterdam.
- [9] Kracht, M. 1994. ‘Logic and Control: How they determine the behavior of presuppositions’. In van Eijk and Visser, eds. *Logic and Information Flow*. MIT Press.
- [10] Peters, S. 1975. ‘A Truth-Conditional Formulation of Karttunen’s Account of Presuppositions’. In *Texas Linguistic Forum* 6.
- [11] Schlenker, P. 2008. ‘Be Articulate! A Pragmatic Theory of Prsupposition Projection’. To appear in *Theoretical Linguistics*.

---

possibility.

- [12] Soames, S. 1989. 'Presupposition'. *Handbook of Philosophical Logic IV* (ed. Gabbay and Guentner).
- [13] Stalnaker, R. 1974. 'Pragmatic presuppositions'. In Milton Munitz and Peter Unger, editors, *Semantics and Philosophy*. New York University Press.