Towards an explanatory theory of presupposition projection

Patrick D. Elliott & Danny Fox

October 9, 2020

homework

- If you have time, read Fox (2013) in advance of next week.
- There is no p-set this week, so use this time to catch up on any material you may have missed.

1 *Recap*

Where we're at

- Last week, we encountered a new perspective on linguistic meanings — *dynamic semantics* (Heim 1982, 1983, Veltman 1996) — wherein the meaning of an expression is an *instruction* to update an information state.
- This week we'll begin by briefly recapping what we discovered, before finishing our discussion of Veltman's semantics for epistemic modals.
- This will conclude our initial encounter with dynamic semantics — we'll return to this topic in a couple of weeks time, when we discuss *anaphora* in dynamic semantics.
- In the mean time, we'll discuss a notable attempt to make the theory of presupposition projection more explanatory by providing a *general algorithm* for deriving trivalent semantics for logical connectives and other operators (George 2008, 2014).

What we've accomplished so far

- We began with a fairly austere trivalent semantics, which gave us a notion of the *semantic presupposition* of a sentence.
- We paired our semantics with a Stalnakerian pragmatics, wherein to *assert* a sentence is to propose updating an idealized body of information, the *context set*.

- The *bridge principle* allowed us to model the interaction between the semantic presuppositions of sentences, and the pragmatics of assertion.
- This theory allowed us to model the interaction between presupposition and information growth in discourse.
- We went on to explore a fairly radical generalization of Stalnakerian pragmatics to the semantics proper, *dynamic semantics*:

> "The slogan 'You know the meaning of a sentence if you know the conditions under which it is true', is replaced by this one: 'You know the meaning of a sentence if you know the change it brings about in the information state of anyone who accepts the news conveyed by it'." (Veltman 1996)

- In order to develop an understanding of the basic features of dynamic semantics, we went through the simplest version of the theory: propositional update semantics (Veltman 1996).
- As we saw in class (hopefully, further reinforced by doing the homework), the resulting system in which updates of complex sentences are assembled compositionally, delivered exactly the same results vis á vis informational contribution as a static semantics with a global (i.e., pragmatic) update rule.[1]
- Even before we considered presupposition projection, an advantage of the resulting system, however, is that it allowed us to formulate a notion of informational *redundancy* active not just at the level of discourse, but also inter-sententially. This collapsed cases like the following:

(1) a. It's raining. (In fact) it's raining heavily.
    b. It's raining heavily. # (In fact) it's raining.

(2) a. It's raining and (in fact) it's raining heavily.
    b. # It's raining heavily and (in fact) it's raining.

(3) a. If it's raining then it's raining heavily.
    b. # If it's raining heavily then it's raining.

- We went on to show that, once we incorporate a notion of *semantic presupposition* into propositional update semantics, along with the bridge principle, the resulting system captures the Karttunen-Heim projection generalizations for the logical connectives.

(4) Sarah has a corgi, and her corgi is cute.

(5) If Sarah has a corgi, then her corgi is cute.

(6) Either Sarah has no corgi, or her corgi is cute.

- We subsequently realized that we shouldn't get *too* excited about these results. As pointed out by Rooth and Soames, update semantics for the logical connectives are essentially *tailored* to derive the Karttunen-Heim generalizations, so it's not surprising that they indeed do so. Update semantics doesn't straightforwardly follow from a classical semantics.[2]

[1] Veltman (1996) demonstrated this by defining an operation of *staticization*.

[2] This is something that should have been reinforced by the homework exercises last week.

- That said, I suggested that we shouldn't hastily dismiss dynamic semantics completely. Update semantics doesn't just capture presupposition projection, but lumps it in with other phenomena which bolster the explanatory power of dynamic semantics.

- One empirical domain, which we'll come back to in a couple of weeks time: *anaphora* (Heim 1982, Groenendijk & Stokhof 1991).

(7) a. Sarah has a^1 corgi and she adores it$_1$.
    b. Everyone who has a^1 puppy showers it$_1$ with affection.

- Another empirical domain, which we started to discuss last week, is *epistemic modality*, and specifically *epistemic contradictions* (Veltman 1996, Yalcin 2007).

(8) a. # It's raining and it might not be raining.
    b. ? It might be raining and it's not raining.

- Before getting back to epistemic modals, we'll provide a brief technical summary of update semantics.

Technical summary of update semantics

- In propositional update semantics with trivalence, a model is a pair $\langle W, I \rangle$, where W is a finite non-empty set of possible worlds, and $I : \mathscr{A} \mapsto (W \mapsto \{1, 0, \#\})$ is an evaluation function.

- Giving an update semantics for a simple propositional language $\mathscr{L}$ consists of recursively defining an update function .[.] mapping *information states* and expressions of $\mathscr{L}$ to (potentially modified) information states.

Atomic sentences

Update of a context c with an atomic sentence p is subject to Stalnaker's bridge; update is defined iff c entails the presupposition of p.

$$c[p] := \begin{cases} c \cap \{ w \mid I(p)\, w \} & c \subseteq p^{\pi} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Negated sentences

Updating c with a negated sentence $\neg \phi$, involves first updating c with ϕ, and subtracting the result from c. This predicts that $\neg \phi$ inherits the presuppositions of ϕ.

$$c[\neg \phi] := c - c[\phi]$$

Conjunctive sentences

Updating c with a conjunctive sentence is an instruction to perform a successive update; this predicts that the presuppositions of ψ are satisfied if entailed by c updated with the first conjunct.

$$c[\phi \wedge \psi] \coloneqq c[\phi][\psi]$$

Disjunctive sentences

Updating c with a disjunctive sentence is an instruction to update c with ϕ, then update c with $\neg\phi \wedge \psi$, and then take the union of the results; this predicts that the presuppositions of ψ are satisifed if entailed by c updated with the *negation* of the first disjunct.

$$c[\phi \wedge \psi] \coloneqq c[\phi] \cup c[\neg\phi][\psi]$$

Conditional sentences

Updating c with a conditional sentence $\phi \rightarrow \psi$ is an instruction to update c with $\phi \wedge \psi$, and subtract the result from $c[\phi]$. The result is then subtracted from c. This predicts that the presuppositions of the consequent ψ are satisfied if entailed by c updated with the antecedent.

$$c[\phi \rightarrow \psi] \coloneqq c - (c[\phi] - c[\phi][\psi])$$

- Epistemic modals are an interesting case of operators that are *only statable* within update semantics; as you will have learned from doing the homework, staticizing $\Diamond$ results in a tautology.

Veltman's epistemic *might*

Epistemic *might* tests whether updating c with the prejacent results in a non-absurd information state.

$$c[\Diamond\phi] \coloneqq \begin{cases} c & c[\phi] \neq \varnothing \\ \varnothing & \text{otherwise} \end{cases}$$

Epistemic *must*

Epistemic *must* tests whether updating c with the negation of the prejacent results in an absurd information state.

$$c[\Box\phi] \coloneqq \begin{cases} \varnothing & c[\neg\phi] \neq \varnothing \\ c & \text{otherwise} \end{cases}$$

1.1 Back to epistemic contradictions

Case 1

(9) It's not raining outside, but it might be raining outside. $\neg p \wedge \Diamond p$

Consistency in update semantics

A sentence ϕ is consistent with respect to c, if $c[\phi] \neq \varnothing$; a sentence ϕ is *consistent* simpliciter, if there is some information state c', s.t., $c'[\phi]$ is consistent.

A good result: (9) is *inconsistent*.

Before giving an informal proof, the intuition is as follows: updating an information state with the information that it's not raining is guaranteed to make a tentative update of "it's raining" fail.

- $c[\neg p \wedge \Diamond p] = c[\neg p][\Diamond p]$
- $= (c - I(p))[\Diamond p]$
- $= \begin{cases} c - I(p) & (c - I(p)[p] \neq \varnothing \\ \varnothing & \text{otherwise} \end{cases}$
- $= \begin{cases} c - I(p) & ((c - I(p) \cap I(p)) \neq \varnothing \\ \varnothing & \text{otherwise} \end{cases}$
- $(c - s) \cap s = \varnothing, \forall s$, hence $(\neg p \wedge \Diamond p)$ is *inconsistent*.

Case 2

What about the other ordering (10)? Although we didn't assign this a # diacritic, arguably there is something deviant about this sentence.

(10) ?It might be raining and it's not raining. $\Diamond p \wedge \neg p$

We can make sense of this in update semantics by using the notion of *coherence*, which we define in terms of a derivative notion of *support*.

Support in update semantics

An information state c *supports* a sentence ϕ iff:

$$c[\phi] = c$$

Other terms which are often used to mean the same thing: c *accepts* ϕ, c *incorporates* ϕ, ϕ *is redundant in* c.

Coherence in update semantics

ϕ is coherent iff there is some non-absurd information state c, s.t., c *supports* ϕ.

Note that *coherence* implies *consistency*: if a non-absurd c supports ϕ, then $c[\phi]$ is consistent, and hence ϕ is consistent simpliciter.

Now we can ask ourselves, is (10) consistent/coherent?

- $c[\Diamond\, p \wedge \neg\, p] = c[\Diamond\, p][\neg\, p]$
- $= \begin{cases} c[\neg\, p] & c[p] \neq \varnothing \\ \varnothing[\neg\, p] & \text{otherwise} \end{cases}$
- $= \begin{cases} c - c[p] & c[p] \neq \varnothing \\ \varnothing & \text{otherwise} \end{cases}$

(10) is *consistent*, since as long as both p and $\neg\, p$ are c-consistent, then updating c with (10) will result in a non-absurd information state — namely, one that supports $\neg\, p$.

Now we can ask, is (10) *coherent*? The answer is no. For the test imposed by $\Diamond\, p$ to be successful in c, c cannot support $\neg\, p$, and for c to support $p \wedge q$, $c[p]$ must support q.[3]

The thought here is that a speaker can only sincerely utter a sentence if it is coherent, and this follows from the fact that a speaker can only sincerely utter a sentence that is redundant with respect to her *her own* information state (see Groenendijk, Stokhof & Veltman 1996 for discussion).

[3] A sketch of a proof by contradiction:

- If $\Diamond\, p \wedge \neg\, p$ is coherent; there exists a c, s.t., $c[\Diamond\, p \wedge \neg\, p] = c$.
- If $c[\Diamond\, p \wedge \neg\, p] = c$, then $(c[\Diamond\, p])[\neg\, p] = c$, so by eliminativity $c[\Diamond p] = c[\neg\, p] = c$
- if $c[\Diamond p] = c$, then $c[p] \neq \varnothing$
- if $c[p] \neq \varnothing$, then $c - p \neq c$
- Therefore $c[\neg\, p] \neq c$

Optional exercise

Recall that, due to presupposition projection facts, the update rule for disjunctive sentences is as follows:

$$c[\phi \vee \psi] \coloneqq c[\phi] \cup c[\neg \phi][\psi]$$

What does the theory predict for a sentence such as "either it's raining, or it might be raining"?

$$p \vee \Diamond p$$

What about the reverse order, "it might be raining, or it's raining"?

$$\Diamond p \vee p$$

Try to connect the results to your intuitions about what these sentences mean.

2 *Towards an explanatory theory of presupposition projection*

2.1 *Back to the Rooth-Soames objection and Schlenker's challenge*

It's advantages in other domains nonwithstanding, there is something stipulative about update semantics in the domain of presupposition projection.

Can we improve upon it? In recent years there have been a number of attempts to cover the same empirical ground as dynamic semantics (wrt presupposition projection) with fewer stipulations, by developing a general *algorithm* which predicts the behavior of bivalent operators in a setting with presuppositions. This point has been made especially forcefully in the work of Philippe Schlenker (2008, 2009, 2010).

If successful, this counts as an improvement over dynamic semantics, since the algorithm only needs to be stated once.

In order to do this in a way that requires minimal additional machinery, well shift back to a static trivalent setting, and think about the interpretation of the third truth value, following George (2008).

2.2 Weak Kleene

Once we introduce a third truth-value (#) there is an extremely large number of ways in which we might consider extending the classical semantics of the logical connectives.

Consider, e.g., conjunction.

$\phi \wedge \psi$

ϕ \ ψ	1	0	#
1	1	0	?
0	0	0	?
#	?	?	?

Once we introduce # as a possible truth-value, there are 5 new cells in the truth table, each of which can have three values, which means that there are 243 (3^5) possible ways of extending conjunction in a trivalent setting.

Of course, not all of these possibilities are going to be useful for analyzing natural language.

Weak Kleene algorithm

Where the classical semantics is silent, always return #.

One way of thinking of the third truth value, #, is as representing *undefinedness*.

This interpretation gives rise to a Weak Kleene logic.

Weak Kleene truth-tables

ϕ	$\neg\phi$
1	0
0	1
#	#

Figure 1: Negated formulas

$\phi \wedge \psi$

ϕ \\ ψ	1	0	#
1	1	0	#
0	0	0	#
#	#	#	#

Figure 2: Conjunctive formulas

$\phi \vee \psi$

ϕ \\ ψ	1	0	#
1	1	1	#
0	1	0	#
#	#	#	#

Figure 3: Disjunctive formulas

$\phi \rightarrow \psi$

ϕ \\ ψ	1	0	#
1	1	0	#
0	1	1	#
#	#	#	#

Figure 4: Conditional formulas

As we've already seen, this isn't much use for analyzing presupposition projection in natural language.

2.3 *Strong Kleene (symmetric)*

We can work towards a more explanatory set of trivalent meanings by shifting perspectives on what the third truth-value # is taken to represent.

We can think of the third truth value, #, as representing *uncertainty whether 1 or 0*, which we can represent as the set $\{1, 0\}$.

In order to explain the recipe, it will be helpful to think of our three truth-values as the following isomorphic three-membered set: $\{\{1\}, \{0\}, \{1, 0\}\}$, with $\{1\}$ representing *definitely true*, $\{0\}$ representing *definitiely false*, and $\{1, 0\}$ representing *maybe true and maybe false*.

$$\{\overbrace{\{1\}}^{\text{true}}, \underbrace{\{0\}}_{\text{false}}, \overbrace{\{1, 0\}}^{\text{uncertain}}\}$$

Our recipe will consist of the following recursive procedure:[4]

[4] This is delivers equivalent results to the initial version of George's (2008) *function deployment*.

- Given a complex formula with an n-place truth-functional connective f, $\ulcorner f\ \phi_1 \ldots \phi_n \urcorner$.
- Assuming that I gives the bivalent interpretation of f as a function, compute $\{\ I(f)\ t_1 \ldots t_n \mid t_1 \in [\![\phi_1]\!]^{tri}, \ldots, t_n \in [\![\phi_n]\!]^{tri}\ \}$.
- The result is the value of $[\![f\ \phi_1 \ldots \phi_n]\!]^{tri}$

Applying the Strong Kleene algorithm to conjunction

When the values of the arguments of the connective are $\{ 1 \}$ or $\{ 0 \}$, the algorithm will simply deliver the classical semantics. We can illustrate this with conjunction.

$$[\![p \wedge q]\!]^{tri} = \{\ t \wedge u \mid t \in I(p) \wedge u \in I(q)\ \}$$

If $I(p)$ and $I(q)$ are singleton sets $\{ t \}$ and $\{ u \}$, this will obviously be equivalent to the classical semantics:

$$= \{\ t \wedge u\ \}$$

What if $I(p)$ is $\{ 0, 1 \}$? The value of the conjunctive formula will differ depending on whether $I(q)$ is $\{ 1 \}$ or $\{ 0 \}$. Assuming that $I(q) = \{ 1 \}$:

(11) $I(p) = \{ 0, 1 \}, I(q) = 1$
  a. $[\![p \wedge q]\!]^{tri} = \{\ t \wedge u \mid t \in \{ 0, 1 \} \wedge u \in \{ 1 \}\ \}$
  b. $= \{\ t \wedge 1 \mid t \in \{ 0, 1 \}\ \}$
  c. $= \{ 0, 1 \}$

Next, assume that $I(q)$ is $\{ 0 \}$.
(12) $I(p) = \{ 0, 1 \}, I(q) = \{ 0 \}$
  a. $[\![p \wedge q]\!]^{tri} = \{\ t \wedge u \mid t \in \{ 0, 1 \} \wedge u \in \{ 0 \}\ \}$
  b. $= \{\ t \wedge 0 \mid t \in \{ 0, 1 \}\ \}$
  c. $= \{ 0 \}$

Since the algorithm is completely symmetric, the same reasoning applies if $I(q)$ is $\{ 0, 1 \}$.

Applying this algorithm to every different combination of truth-values, we end up with the following semantics for strong Kleene disjunction. The intuition here is that uncertainty only projects when indeterminacy could affect the result of the conjunctive sentence.

Strong Kleene conjunction

- $\ulcorner \phi \wedge \psi \urcorner$ is *defined* if either (a) $[\![\phi]\!]$ is false, (b) $[\![\psi]\!]$ is false, or (c) both $[\![\phi]\!]$ and $[\![\phi]\!]$ are true.
- $\ulcorner \phi \wedge \psi \urcorner$ is *true* if both $[\![\phi]\!]$ and $[\![\psi]\!]$ are true.
- $\ulcorner \phi \wedge \psi \urcorner$ is *false* if either (a) $[\![\phi]\!]$ is false, or (b) $[\![\psi]\!]$ is false.

Applying the strong Kleene algorithm to disjunction

Again, when the values of the arguments of the connective are $\{ 1 \}$ or $\{ 0 \}$, the algorithm will simply deliver the classical semantics. This is easy to see.

$$[\![p \vee q]\!]^{tri} = \{ t \vee u \mid t \in I(p) \wedge u \in I(q) \}$$

What if $I(p)$ is $\{ 0, 1 \}$? As before, the value of the disjunctive formula will differ deppending on whether $I(q)$ is $\{ 1 \}$ or $\{ 0 \}$. Assuming that $I(q) = 1$:

(13) a. $I(p) = \{ 0, 1 \}, I(q) = \{ 1 \}$
b. $[\![p \vee q]\!]^{tri} = \{ t \vee u \mid t \in \{ 0, 1 \} \wedge u \in \{ 1 \} \}$
c. $= \{ t \vee 1 \mid t \in \{ 0, 1 \} \}$
d. $= \{ 1 \}$

Next, assume that $I(q)$ is $\{ 0 \}$.

(14) a. $I(p) = \{ 0, 1 \}, I(q) = \{ 0 \}$
b. $[\![p \vee q]\!]^{tri} = \{ t \vee u \mid t \in \{ 0, 1 \} \wedge u \in \{ 0 \} \}$
c. $= \{ t \vee 0 \mid t \in \{ 0, 1 \} \}$
d. $= \{ 0, 1 \}$

Like before, since the algorithm is completely symmetric, the same reasoning applies if $I(q)$ is $\{ 0, 1 \}$

Applying the algorithm to every possible combination of truth values, we end up with the following strong Kleene semantics for disjunction. Remember, the intuition is that uncertainty projects when indeterminacy could effect the result of the disjunctive sentence.

Strong Kleene disjunction

- $\ulcorner \phi \vee \psi \urcorner$ is *defined* if either (a) $[\![\phi]\!]$ is true, (b) $[\![\psi]\!]$ is true, or (c) both $[\![\phi]\!]$ and $[\![\phi]\!]$ are false.
- $\ulcorner \phi \vee \psi \urcorner$ is *false* if both $[\![\phi]\!]$ and $[\![\psi]\!]$ are false.
- $\ulcorner \phi \vee \psi \urcorner$ is *true* if either (a) $[\![\phi]\!]$ is true, or (b) $[\![\psi]\!]$ is true.

2.4 *Strong Kleene truth-tables*

Applying the strong Kleene algorithm to the classical connectives, substituting in $\{1, 0, \#\}$ for $\{\{1\}, \{0\}, \{1, 0\}\}$, the result is the following truth-tables.

Strong Kleene truth-tables

The highlighted cells differ from weak Kleene.

ϕ	$\neg\phi$
1	0
0	1
#	#

Figure 5: Negated formulas

$\phi \wedge \psi$

$\phi \backslash \psi$	1	0	#
1	1	0	#
0	0	0	0
#	#	0	#

Figure 6: Conjunctive formulas

$\phi \vee \psi$

$\phi \backslash \psi$	1	0	#
1	1	1	1
0	1	0	#
#	1	#	#

Figure 7: Disjunctive formulas

$\phi \rightarrow \psi$

$\phi \backslash \psi$	1	0	#
1	1	0	#
0	1	1	1
#	1	#	#

Figure 8: Conditional formulas

2.5 *The role of linear order in presupposition projection*

As discussed by Schlenker (2008), it's not clear that the projection generalization we've been assuming for disjunctive sentences is correct.

(15) a. Either this house has no bathroom, or the bathroom is upstairs.
b. Either the bathroom is upstairs, or (else) this house has no bathroom.

(Schlenker 2008: p. 185)

The entry we gave for disjunction in propositional update semantics (after Beaver 2001) can capture the projection pattern illustrated by (15a), but not (15b).

This is because, in update semantics, when we compute $c[\phi \vee \psi]$, the first disjunct ϕ updates the global input context c, but the second disjunct updates a modified context $c[\neg\, \phi]$.[5]

Schlenker suggests that there is a similar problem involving conditional sentences.

(16) a. If this house has a bathroom, then the bathroom is well hidden.
b. If the bathroom is well hidden, then this house has a bathroom.

(Schlenker 2008: p. 186)

He furthermore suggests that post-posing the antecedent makes no difference to the judgements.

(17) a. The bathroom is well hidden, if this house has a bathroom.
b. Mary's doctor knows she is expecting a child, if she is pregnant.

(Schlenker 2008: p. 186)

Nonetheless, it seems like a pretty bad result if we predict that the following presupposes that *if Sarah was sad yesterday, then Sarah has a corgi*.

(18) Sarah's corgi cheered her up today and she was sad yesterday.

We'll put the data favouring strong Kleene to one side for now, and explore the possibility that we can *incrementalize* the strong Kleene algorithm in order to capture the Karttunen-Heim projection generalizations.[6]

3 *Incrementalizing Strong Kleene: towards Middle Kleene*

As we've seen, presupposition projection displays *asymmetries*; something that strong Kleene fails to capture.

We need to adjust the strong Kleene algorithm to account for ordering asymmetries.

[5] As a reminder, here's the semantics for disjunctive formulas in propositional update semantics:

- $c[\phi \vee \psi] = c[\phi] \cup c[\neg\, \phi][\psi]$

[6] This strategy was pursued originally by Schlenker (2008), who proposes a very different kind of algorithmic procedure. Our discussion here is based primarily of George's (2007, 2008, 2014) related work.

As before, it will be helpful to think of our three truth-values as the following isomorphic three-membered set: $\{ \{ 1 \}, \{ 0 \}, \{ 1, 0 \} \}$, with $\{ 1 \}$ representing *definitely true*, $\{ 0 \}$ representing *definitiely false*, and $\{ 1, 0 \}$ representing *maybe true and maybe false*.

The intuition behind our recipe will be a recursive procedure:

Step 1: Let f be an n-place curried function which returns truth-values, X be an n-long sequence of arguments (where $n \geq 1$).

Step 2: If $X \coloneqq [x_1]$, compute $\{ f\,t_1 \mid t_1 \in [\![x_1]\!]^{tri} \}$ and return the result.

Step 3: Else, if $X \coloneqq [x_1, x_2, \ldots]$, compute $\{ f\,t_1 \mid t_1 \in [\![x_1]\!]^{tri} \}$.

- If the result is non-singleton set of functions, return $\{ 0, 1 \}$ as the value of $f\ [\![x_1]\!]^{tri} \ldots [\![x_n]\!]^{tri}$.
- If the result is a singleton set $\{ g \}$, go back to step 1 and let f be g, and $X \coloneqq [x_2, \ldots]$.

3.1 *Applying the Middle Kleene algorithm to conjunction*

Case 1: Presuppositions in the first conjunct

Let's start by applying the algorithm to a conjunctive formula $p \wedge q$, where $I(p) = \{ 0, 1 \}$, and $I(q) = \{ 0 \}$.

By assumption, we have a curried function $\lambda t\ .\ \lambda u\ .\ t \wedge u$, and a sequence of arguments $[p, q]$.

We begin by applying our function pointwise to each value in $I(p)$.

$$\begin{aligned} &\{ [\lambda t\ .\ \lambda u\ .\ t \wedge u]\ t \mid t \in \{ 1, 0 \} \} \\ &= \{ \lambda u\ .\ 1 \wedge u, \lambda u\ .\ 0 \wedge u \} \end{aligned}$$

These are *distinct functions*, which we can see by looking at their graphs, this means that we get back $\{ 1, 0 \}$ (i.e. $\#$) as the value of $[\![p \wedge q]\!]$.

$$[\lambda u\ .\ 1 \wedge u] \coloneqq \begin{bmatrix} \mathbf{1 \mapsto 1} \\ 0 \mapsto 0 \end{bmatrix}$$

$$[\lambda u \,.\, 0 \wedge u] := \begin{bmatrix} \mathbf{1} \mapsto \mathbf{0} \\ 0 \mapsto 0 \end{bmatrix}$$

The intuition here is that, as soon as the function is fed an argument that *could* lead to indeterminacy, the algorithm throws its hands up and returns a presupposition failure.

Note that the value of $I(q)$ *didn't in fact matter* — since the value of feeding in the first conjunct was indeterminate, the algorithm simply throws up its hands and returns $\{ 1, 0 \}$.

Case 2: Presuppositions in the second conjunct

Now let's apply the algorithm to a conjunctive formula $p \wedge q$, where $I(p) = \{ 0 \}$ and $I(q) = \{ 1, 0 \}$.

By assumption, we have a curried function $\lambda t \,.\, \lambda u \,.\, t \wedge u$ and a sequence of arguments $[p, q]$.

We begin by applying our function pointwise to each value in $I(p)$; since $I(p)$ is the singleton set $\{ 0 \}$, this is straightforward:

$$\begin{aligned} &\{ [\lambda t \,.\, \lambda u \,.\, t \wedge u]\, t \mid t \in \{ 0 \} \} \\ &= \{ \lambda u \,.\, 0 \wedge u \} \end{aligned}$$

The result is a singleton set containing a function, so we can go back and repeat our algorithm. We apply the function pointwise to each value in $I(q)$:

$$\begin{aligned} &\{ [\lambda u \,.\, 0 \wedge u]\, u \mid u \in \{ 1, 0 \} \} \\ &= \{ 0 \wedge 1, 0 \wedge 0 \} \\ &= \{ 0, 0 \} = \{ 0 \} \end{aligned}$$

The result is a singleton set containing a truth-value, which gives us the semantic value of the conjunctive sentence.

If the second conjunct had been $\{ 1, 0 \}$ of course, we would have gotten back a different result — namely, $\{ 1, 0 \}$. When the third value occurs in the *second* conjunct, middle Kleene algorithm therefore does the same thing as strong Kleene.

3.2 *Applying the middle Kleene algorithm to disjunction*

Case 1: Presuppositions in the first disjunct

Let's start by applying the algorithm to a disjunctive formula $p \vee q$, where $\boldsymbol{I}(p) = \{ 1, 0 \}$, and $\boldsymbol{I}(q) = \{ 1 \}$.

By assumption, we have a curried function $\lambda t \,.\, \lambda u \,.\, t \vee u$, and a sequence of arguments $[p, q]$.

We begin by applying our function pointwise to each value in $\boldsymbol{I}(p)$.

$$\begin{aligned} &\{ [\lambda t \,.\, \lambda u \,.\, t \vee u]\, t \mid t \in \{ 1, 0 \} \} \\ &= \{ \lambda u \,.\, 1 \vee u, \lambda u \,.\, 0 \vee u \} \end{aligned}$$

These are *distinct functions*.

$$[\lambda u \,.\, 1 \vee u] \coloneqq \begin{bmatrix} 1 \mapsto 1 \\ 0 \mapsto 1 \end{bmatrix}$$

$$[\lambda u \,.\, 0 \wedge u] \coloneqq \begin{bmatrix} 1 \mapsto 1 \\ 0 \mapsto 0 \end{bmatrix}$$

Case 2: Presuppositions in the second disjunct

Now let's apply the algorithm to a disjunctive formula $p \vee q$, where $\boldsymbol{I}(p) \coloneqq \{ 1 \}$, and $\boldsymbol{I}(q) = \{ 1, 0 \}$.

By assumption, we have a curried function $\lambda t \,.\, \lambda u \,.\, t \vee u$, and a sequence of arguments $[p, q]$.

We begin by applying our function pointwise to each value in $\boldsymbol{I}(p)$.

$$\begin{aligned} &\{ [\lambda t \,.\, \lambda u \,.\, t \vee u]\, t \mid t \in \{ 1 \} \} \\ &= \{ \lambda u \,.\, 1 \vee u \} \end{aligned}$$

The result is determinate, so in accordance with the algorithm we take the result and apply it pointwise to the value of the next argument:

$$\{\,[\lambda u\,.\,1 \vee u]\,u \mid u \in \{\,1,0\,\}\,\}$$
$$= \{\,1 \vee 1, 1 \vee 0\,\}$$
$$= \{\,1 \vee 1\,\} = \{\,1\,\}$$

This gives us the semantic value of the sentence as $\{\,1\,\}$.

Note that things would have been different had the value of p been $\{\,0\,\}$. In this instance, subsequent pointwise application would have failed to collapse the result, and we would have ended up with $\{\,1,0\,\}$. When the third value occurs in the *second* disjunct, the middle Kleene algorithm therefore does the same thing as strong Kleene.

3.3 Assembling the results

Middle Kleene truth-tables

N.b. the highlighted cells diverge from strong Kleene.

ϕ	$\neg\phi$
1	0
0	1
#	#

Figure 9: Negated formulas

$\phi \wedge \psi$

ϕ \ ψ	1	0	#
1	1	0	#
0	0	0	0
#	#	#	#

Figure 10: Conjunctive formulas

$\phi \vee \psi$

ϕ \ ψ	1	0	#
1	1	1	1
0	1	0	#
#	#	#	#

Figure 11: Disjunctive formulas

$\phi \to \psi$

ϕ \ ψ	1	0	#
1	1	0	#
0	1	1	1
#	#	#	#

Figure 12: Conditional formulas

3.4 Equivalence with propositional update semantics

To show that middle Kleene delivers the same results as update semantics for presupposition projection, it's helpful to be a little more explicit about the pragmatic component factors in.

We'll again assume a simple propositional language $\mathscr{L}$ interpreted relative to a model $\langle W, I\rangle$, where W is a finite non-empty set of possible worlds and I maps atomic sentences to partial propositions. Since our semantics is *static* however, we'll recursively define $[\![.]\!]$ as a function from any sentence in $\mathscr{L}$ to a partial proposition.

Connectives in the language will simply be interpreted as their *middle Kleene* counterparts (I'll write, e.g., $\wedge_{mid}$ for the middle Kleene entry for conjunction).

Definition 3.1 (Simple sentences)**.**

$$[\![p]\!] \coloneqq I(p)$$

Definition 3.2 (Conjunctive sentences)**.**

$$[\![p \wedge q]\!] \coloneqq \lambda w \,.\, I(p)(w) \wedge_{mid} I(q)(w)$$

Definition 3.3 (Disjunctive sentences)**.**

$$[\![p \vee q]\!] \coloneqq \lambda w \,.\, I(p)(w) \vee_{mid} I(q)(w)$$

Definition 3.4 (Conditional sentences)**.**

$$[\![p \rightarrow q]\!] \coloneqq \lambda w \,.\, I(p)(w) \rightarrow_{mid} I(q)(w)$$

Finally, as a seperate component, we give our notion of *update* — since this semantics is *static* update is tied to the pragmatics of assertion.

Definition 3.5 (Update)**.**

$$c[\phi] \coloneqq \begin{cases} \{\, w \mid w \in c \wedge [\![\phi]\!]\ w = 1 \,\} & \forall w \in c[[\![\phi]\!]\ w = 1 \vee [\![\phi]\!]\ w = 0] \\ \text{undefined} & \text{otherwise} \end{cases}$$

Now we'll give a couple of illustrations to show that this system delivers the same results as propositional update semantics.

Example 1: Projection in conditional sentences

Let's go through a concrete concrete case, in w_{cy} Sarah has a corgi and it's cute, in w_{cn} Sarah has a corgi and it's not cute, and in $w_{\varnothing}$ Sarah has no corgi.

(19) If Sarah has a corgi, then Sarah's corgi is cute. $p \to q$

$\{ w_{cy}, w_{cn}, w_{\varnothing} \} [p \to q]$ is defined if $\{ w_{cy}, w_{cn}, w_{\varnothing} \}$ entails the semantic presupposition of $p \to q$.

$\phi \to \psi$

ϕ \ ψ	1	0	#
1	1	0	#
0	1	1	1
#	#	#	#

Figure 13: Middle Kleene semantics for material implication

- According to the middle Kleene truth-table, $p \to q$ is true at a world w if (a) Sarah has no corgi in w, or (b) Sarah has a corgi in w and Sarah's corgi is cute in w. These worlds are $\{ w_{cy}, w_{\varnothing} \}$.
- $p \to q$ is false at a world w if Sarah has a corgi in w, and Sarah's corgi is not cute in w. These worlds are $\{ w_{cn} \}$.

The semantic presupposition of the sentence is therefore just $\{ w_{cy}, w_{cn}, w_{\varnothing} \}$, which is entailed (and in fact equivalent to) the context set c.

Since update is defined, we simply retain the *true* worlds from the context set, which are $\{ w_{cy}, w_{\varnothing} \}$.

Let's check that we get the desired results if we flip the order of antecedent and consequent.

(20) If Sarah's corgi is cute, then Sarah has a corgi. $q \to p$

Now we compute the semantic presupposition of the sentence:

- According to the middle Kleene truth-table, $q \to p$ is true at a world w if (a) q is false in w, or (b) q and p are true in w. In other words, either Sarah has a corgi and it's not cute, or Sarah has a corgi and it's cute. These worlds are $\{ w_{cy}, w_{cn} \}$.
- $q \to p$ is false at a world iff q is true and p is false. I.e., Sarah has a corgi and it's cute, and Sarah doesn't have a corgi. This is a contradiction, so these worlds are $\varnothing$.

The semantic presupposition of this sentence is therefore $\{ w_{cy}, w_{cn} \}$, i.e., *that Sarah has a corgi*, and update will be undefined in the context above.[7]

[7] A symmetric theory would predict this sentence to be presuppositionless, although there's an independent factor which could be responsible for the oddness of (20); namely, if we incorporate some notion of incremental redundancy into our theory, the consequent is predicted to be redundant.

Example 2: Projection in disjunctive sentences

Again, assume w_{cy} Sarah has a corgi and it's cute, in w_{cn} Sarah has a corgi and it's not cute, and in $w_{\varnothing}$ Sarah has no corgi.

(21) Either Sarah has no corgi, or Sarah's corgi is cute. $\neg\, p \vee q$

$\{\, w_{cy}, w_{cn}, w_{\varnothing} \,\}\, [p \vee q]$ is defined if $\{\, w_{cy}, w_{cn}, w_{\varnothing} \,\}$ entails the semantic presupposition of $\neg\, p \vee q$.

$\phi \vee \psi$

ϕ \ ψ	1	0	#
1	1	1	1
0	1	0	#
#	#	#	#

Figure 14: Middle Kleene semantics for disjunction

- According to the middle Kleene truth-tables $\neg\, p \vee q$ is true at a world w if (a) p is false in w, or (b) p is true in w and q is true in w. In other words, worlds in which Sarah doesn't have a corgi, and those in which she does, and it's cute. These are $\{\, w_{\varnothing}, w_{cy} \,\}$.
- $\neg\, p \vee q$ is false at a world w if p is true and q is false; so, if Sarah has a corgi, and it's not cute. These are just $\{\, w_{cn} \,\}$.
- The semantic presupposition, then is $\{\, w_{cy}, w_{cn}, w_{\varnothing} \,\}$, which is entailed by (in fact equivalent to) the original context.

Since the update is defined, we now simply retain the true worlds, which are $\{\, w_{\varnothing}, w_{cy} \,\}$.

Optional exercise

Show how the predictions change if the order of the disjuncts is flipped.

4 *Extension to first order quantifiers*

The empirical issue

Many theories of projection, such as Heim's system, predict quantified sentences to have universal presuppositions *across the board*.[8]

[8] We haven't actually seen this yet; we'll take more about how dynamic semantics deals with quantificational sentences once we talk about anaphora.

This seems plausible in many cases:

(22) a. Each of these Italians has stopped smoking.
presupposes: each Italian smoked in the past
b. None of these Italians has stopped smoking.
presupposes: each Italian smoked in the past

For other quantifiers however, this is much less straightforward; they seem to presuppose something much weaker.

(23) a. At least one Italian has stopped smoking.
presupposes: some Italian smoked in the past
b. Some Italian has stopped smoking.
presupposes: some Italian smoked in the past

Here we'll abstract away from the intricacies of the data (but more on this from Danny next week), and show how a generalization of the strong Kleene algorithm predicts a contrast between first order existential and universal quantifiers.

Generalizing the strong Kleene algorithm

Consider a language $\mathscr{L}'$ just like $\mathscr{L}$, only we supplement it with a non-empty finite set of *predicate symbols* $\mathscr{P} = \{ P, Q, ... \}$, and quantifiers $\ulcorner\exists\urcorner$, $\ulcorner\forall\urcorner$, s.t., if $P \in \mathscr{P}$ then $\ulcorner\exists\ P\urcorner, \ulcorner\forall\ P\urcorner \in \mathscr{L}'$.

The evaluation function I maps predicate symbols to total functions from a domain of individuals D to $\{ \{ 1 \}, \{ 0 \}, \{ 1, 0 \} \}$, and the quantifiers to the following functions to bivalent truth-values:

(24) a. $I(\exists) \coloneqq \lambda P_{et}\,.\,\exists x[x \in D \wedge P\ x]$ $\langle \mathtt{et}, \mathtt{t} \rangle$
b. $I(\forall) \coloneqq \lambda P_{et}\,.\,\forall x[x \in D \rightarrow P\ x]$ $\langle \mathtt{et}, \mathtt{t} \rangle$

In a trivalent setting, the scope of a quantifier will be a function from $f\ :\ D \mapsto \{ \{ 1 \}, \{ 0 \}, \{ 1, 0 \} \}$; since we aren't here interested in linear ordering effects, we need a way of generalizing the strong Kleene algorithm to quantifiers.

Definition 4.1 (Semantics of quantified sentences)**.** Where $D \in \{ \exists, \forall \}$:

$$[\![D\ P]\!] \coloneqq \{ I(D)\ g \mid g \in \{ f \mid f\ x \in I(P)\ x \mid x \in D \} \}$$

The informal intuition:

- P denotes a mapping from D to a set of truth-values.

- We compute a set of functions f, where f maps each x to an element of whatever $I(P)$ maps x to.
- This means that if $I(P)$ maps x to a singleton set, we get one f, whereas if $I(P)$ maps x to a 2-membered set, we get two fs.
- The resulting set of mappings from D to $\{1, 0\}$ is applied *pointwise* to the function denoted by the quantified, resulting in a set of truth-values.

Let's say that we're interested in "stopped smoking", which maps individuals to $\{1\}$ if they smoked and don't smoke anymore, $\{0\}$, if they smoked and still smoke, and $\{1, 0\}$ if they never smoked:

- $I(P)(\text{sophie}) = \{1\}$
- $I(P)(\text{paul}) = \{0\}$
- $I(P)(\text{nathan}) = \{1, 0\}$

Case 1: Existential quantification

In order to compute $[\![\exists\ P]\!]$, we first compute $\{ f \mid f\ x \in I(P)\ x \mid x \in D \}$. The result is the following set of functions; $I(P)$ maps everything except Nathan to a singleton set, so functions only differ along the *Nathan* dimension:

$$f_1 := \begin{bmatrix} \text{sophie} \mapsto 1 \\ \text{paul} \mapsto 0 \\ \text{nathan} \mapsto 1 \end{bmatrix}$$

$$f_2 := \begin{bmatrix} \text{sophie} \mapsto 1 \\ \text{paul} \mapsto 0 \\ \text{nathan} \mapsto 0 \end{bmatrix}$$

Now we apply $\exists$ pointwise to each of f_1 and f_2.

$$[\![\exists\ P]\!] = \{ [\lambda P\ .\ \exists x \in D[P\ x]]\ f \mid f \in \{ f_1, f_2 \} \}$$

Due to the semantics of existential quantification, the result is true in each case, so the semantic value for $[\![\exists\ P]\!]$ is $\{1\}$.

The prediction in the general case is that an existential sentence will be *true* just if P is true of at least one individual. In other words, we get *existential projection*.

What about if there are no individuals that P is true of? Let's shrink our domain of individuals $D' = \{ \text{paul}, \text{nathan} \}$. Computing $\{ f \mid f\ x \in I(P)\ x \mid x \in D' \}$ gives us the following set of functions:

$$f_1 \coloneqq \begin{bmatrix} \text{paul} \mapsto 0 \\ \text{nathan} \mapsto 1 \end{bmatrix}$$

$$f_2 \coloneqq \begin{bmatrix} \text{paul} \mapsto 0 \\ \text{nathan} \mapsto 0 \end{bmatrix}$$

Applying the existential quantifier pointwise to these functions will result in $\{ 1, 0 \}$. In other words, we predict that an existential sentence is only false if the scope is false of *every* individual; otherwise, it is undefined.

From this discussion, it's easy to see that an existential sentence $\exists\ P$ is predicted to presupposes *either* (a) someone is P, or (b) nobody is P.

Case 2: Universal quantification

Let's tweak our model slightly, and assume that both Paul and Sophie used to smoke but don't anymore, whereas Nathan never smoked.

- $I(P)(\text{paul}) = I(P)(\text{sophie}) = \{ 1 \}$
- $I(P)(\text{nathan}) = \{ 1, 0 \}$

We're interested in the status of the following sentence:

(25) Everyone stopped smoking. $\forall\ P$

We first compute $\{ f \mid f\ x \in I(P)\ x \mid x \in D \}$. The result is the following set of functions:

$$f_1 \coloneqq \begin{bmatrix} \text{paul} \mapsto 1 \\ \text{sophie} \mapsto 1 \\ \text{nathan} \mapsto 1 \end{bmatrix}$$

$$f_2 \coloneqq \begin{bmatrix} \textsf{paul} \mapsto 1 \\ \textsf{sophie} \mapsto 1 \\ \textsf{nathan} \mapsto 0 \end{bmatrix}$$

Applying ∀ pointwise to each function results in $\{0, 1\}$ — i.e., a presupposition failure. For a universal sentence to be true, the scope must be true of *every* individual.

Let's now change our model again, assuming that both Paul and Sophie used to smoke, and still do, whereas Nathan never smoked.

- $\boldsymbol{I}(\boldsymbol{P})(\textsf{paul}) = \boldsymbol{I}(\boldsymbol{P})(\textsf{sophie}) = \{0\}$
- $\boldsymbol{I}(\boldsymbol{P})(\textsf{nathan}) = \{1, 0\}$

$$f_1 \coloneqq \begin{bmatrix} \textsf{paul} \mapsto 0 \\ \textsf{sophie} \mapsto 0 \\ \textsf{nathan} \mapsto 1 \end{bmatrix}$$

$$f_2 \coloneqq \begin{bmatrix} \textsf{paul} \mapsto 0 \\ \textsf{sophie} \mapsto 0 \\ \textsf{nathan} \mapsto 0 \end{bmatrix}$$

Applying ∀ pointwise to each function results in $\{0\}$, i.e., a universal sentence is predicted to be false, just in case the scope is false of some individual.

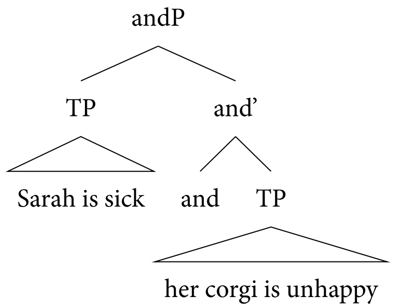
From this discussion, it's easy to see that a universal sentence $\forall\ \boldsymbol{P}$ is predicted to presuppose either (a) everyone is $\boldsymbol{P}$, or (b) at least one person isn't $\boldsymbol{P}$.

5 *Incrementalization: beyond middle Kleene*

We've presented George's middle Kleene algorithm with reference to a toy fragment where issues of compositionality and syntax don't really arise, but there's an open question as to what determines the order in which arguments are evaluated for the Middle Kleene and related algorithms.

Based on syntactic evidence, it's often assumed that conjunctive sentences in English have a descending structure:

(26)



This suggests the following currying:

(27) $[\![\text{and}]\!] \coloneqq \lambda u\ .\ \lambda t\ .\ t \wedge u$

In order to get the projection facts right, we can’t incrementalize our algorithm based on the *evaluation order* provided by the curried function; rather incrementalization must make reference to the order in which the juncts are pronounced.

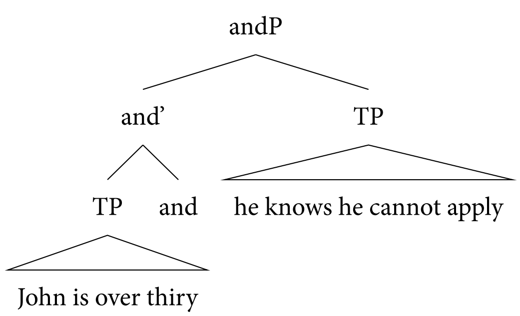
This intuition is bolstered by the fact that in head-final languages such as Korean and Japanese, where the constituency of conjunctive sentences is different, the projection generalizations are the same as in English.

Consider the following Korean example; Chung (2018) observes that the sentence presupposes that *if John is over thirty, he cannot apply*.

(28) *[John-un selun-i nem-ess-ko] caki-ka ciwenha-ci*
[John-TOP thirty-NOM over-perf-and] self-NOM apply-CI
mosha-n-ta-num
cannot-PRES-DECL-RES
“John is over thirty and he knows he cannot apply.”
(Chung 2018: p. 319)

Since this is a head final language, the structure is assumed to be as follows:

(29)



This motivates a different currying of the conjunctive marker:

(30) $[\![ko]\!] \coloneqq \lambda t \,.\, \lambda u \,.\, t \wedge u$

Here, incrementalizing based on evaluation and linear order coincide, but we're clearly missing a generalization if we give different algorithms for English and Korean.

Intriguingly, there is evidence from redundancy effects in Japanese, discussed by Ingason (2016), that suggest that incrementalization is not strictly based on linear order.[9]

[9] We haven't shown here how to give an explanatory theory of redundancy effects, that doesn't stipulate the dynamic entries for the connectives, but see, e.g., Schlenker (2009).

(31)	*Taro-ga*	*[[**yamome**-dearu]*	***zyosei**-ni]*	*atta.*
	Taro-NOM	[[**widow**-COP]	**woman**-DAT]	met.
	"Taro met a woman who is a widow"			

(32) #	*Taro-ga*	*[[**zyosei**-dearu]*	***yamome**-ni]*	*atta.*
	Taro-NOM	[[**woman**-COP]	**widow**-DAT]	met.
	"Taro met a widow who is a woman"			

According to Ingason (2016), the judgements above indicate that a relativve clause is interpreted in the context of the head noun, irregardless of linear order.

There's a common assumption that redundancy and presupposition projection pattern together with respect to incrementality, but this is not obvious at all.[10]

[10] In fact, we already saw one place in which presupposition projection and redundancy apparently come apart — namely, disjunctions (Mayr & Romoli 2016).

The factors that incrementalization is sensitive to is very much an open question.[11]

[11] Indeed, something relating to this question would make for an excellent squib topic.

References

Beaver, David I. 2001. *Presupposition and Assertion in Dynamic Semantics*. 250 pp.

Chung, WooJin. 2018. Context updates in head-final languages: Linear order or hierarchy? *Proceedings of Sinn und Bedeutung* 22(1). 313–329.

Fox, Danny. 2013. Presupposition projection from quantificational sentences - Trivalence, local accommodation, and presupposition strengthening. In Ivano Caponigro & Carlo Cecchetto (eds.), *From grammar to meaning*, 201–232.

George, B. R. 2007. Predicting presupposition projection - Some alternatives in the strong Kleene tradition. unpublished manuscript. UCLA.

George, B. R. 2008. A new predictive theory of presupposition projection. In *Proceedings of SALT 18*, 358–375. Ithaca, NY: Cornell University.

George, B. R. 2014. Some remarks on certain trivalent accounts of presupposition projection. *Journal of Applied Non-Classical Logics* 24(1-2). 86–117.

Groenendijk, J. a. G., M. J. B. Stokhof & F. J. M. M. Veltman. 1996. *Coreference and modality*. OxfordBlackwell.

Groenendijk, Jeroen & Martin Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy* 14(1). 39–100.

Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*. University of Massachusetts - Amherst dissertation.

Heim, Irene. 1983. On the projection problem for presuppositions. In *Proceedings of WCCFL 2*, 114–125. Stanford University.

Ingason, Anton. 2016. Context updates are hierarchical. *Glossa: a journal of general linguistics* 1(1). 37.

Mayr, Clemens & Jacopo Romoli. 2016. A puzzle for theories of redundancy - Exhaustification, incrementality, and the notion of local context. *Semantics and Pragmatics*.

Schlenker, Philippe. 2008. Be Articulate: A pragmatic theory of presupposition projection. *Theoretical Linguistics* 34(3).

Schlenker, Philippe. 2009. Local contexts. *Semantics and Pragmatics* 2.

Schlenker, Philippe. 2010. Local contexts and local meanings. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition* 151(1). 115–142.

Veltman, Frank. 1996. Defaults in Update Semantics. *Journal of Philosophical Logic* 25(3). 221–261.

Yalcin, Seth. 2007. Epistemic Modals. *Mind* 116(464). 983–1026.