Dynamics

Daniel Rothschild and Seth Yalcin*

February 22, 2012

Abstract

Dynamic semantics (as developed by Heim, Kamp, Groenendijk and Stokhof among others) has been one of the most prominent semantic frameworks, particularly known for its elegant treatment of presupposition, anaphora, and epistemic modals. Despite much work on the formal foundations of dynamic semantics, there is as yet no simple characterization of what formal properties make a semantics dynamic rather than static. We extend work by van Bentham and Veltman to give necessary and sufficient conditions for a semantics to be static in the classic, Stalnakerian sense. We then deploy this characterization to assess the empirical case that natural language is dynamic rather than static.

1 The question of static versus dynamic

A familiar and influential picture of linguistic communication, due largely to Stalnaker [1970, 1978], runs as follows: conversation takes place against a *common ground*, a body of information mutually taken for granted, or presupposed, by the discourse participants in context. The common ground is modeled as a set of maximally specific states of affairs, or possible worlds—the possibilities left open by what is mutually presupposed. The meaning of a declarative sentence in context

*We are grateful to Salah Baouendi, Emmanuel Chemla, Benj Hellie, Dilip Ninan, Linda Rothschild, and Lawrence Valby for discussion.

is characterized by the set of possibilities in which it is true—a truth-condition, or *proposition*. Speech acts are modeled from the perspective of their characteristic effect on the common ground. The speech act of assertion in particular is modeled as a 'proposal' to add the proposition asserted to the common ground: when an assertion is successful, the context is updated to rule out all possibilities in which the sentence is false. The set of possibilities compatible with the common ground is intersected with the set of possibilities compatible with the truth of the sentence to yield a new common ground.

We can say that, on Stalnaker's kind of approach, every declarative sentence in context is associated with (i) a truth-condition, and (ii) a *context change potential* (CCP), an operation ('update') on the common ground. (Henceforth we use generally use 'context' as synonymous with 'common ground'.) Sentences are associated with truth-conditions directly, via the semantics of the language. Sentences are associated with context change potentials indirectly, via their truth-conditions together with the intersective update rule which characterizes the pragmatic effect of assertion. To state this more precisely: Let $[\![s]\!]$ denote the semantic value of sentence s (some set of worlds), $[s]$ denote the context change potential of s, and $c[s]$ denote the result of updating c with $[s]$.[1] Then we can think of Stalnaker as defining $[\cdot]$ for arbitrary s in terms of $[\![s]\!]$ as follows:

Stalnaker Update. For all contexts c and sentences s, $c[s] =_{\text{def}} c \cap [\![s]\!]$

It will be useful to characterize Stalnaker's framework as determining a kind of *update system*.

Def 1. An UPDATE SYSTEM is a triple $\langle L, C, [\cdot]\rangle$, where L is a set of sentences, C is the set of contexts, and $[\cdot]$ is a function from L to the set of unary operations on C (i.e., the set of CCPs on C).

Take it as uncontroversial (for now) that any given natural language determines some update system or other.[2] The sort of update system Stalnaker's framework delivers is one that is completely determined by the truth-conditions semantically associated with each sentence. We will say Stalnaker assumes a *static semantics*, and uses it to induce a certain kind of *static* update system.

[1]Following standard practice, we use postfix notation when characterizing CCPs, and take it that $c[s][s'] := (c[s])[s']$.

[2]In fact this assumption conflicts with the standard, Kaplanian view of context-sensitivity, which Stalnaker shares. We return to this issue in 6.

Def 2. A STATIC SEMANTICS is a triple $\langle L, W, [\![\cdot]\!]\rangle$, where L is a set of sentences, W is a set of points, and $[\![\cdot]\!]$ is an interpretation function, with $[\![\cdot]\!] : L \to \mathcal{P}(W)$.

Def 3. An update system $\langle L, C, [\cdot]\rangle$ is STATIC if and only if there exists a static semantics $\langle L, W, [\![\cdot]\!]\rangle$ and a one-to-one function f from C to $\mathcal{P}(W)$ such that for all $c \in C$ and $s \in L$, $f(c) \cap [\![s]\!] = f(c[S])$.

The intuitive idea is that an update system is static if it is representable by a system where (a) contexts are treated as sets of points, (b) sentences are semantically associated with sets of points, and (c) updates are determined intersectively. We define static this way in order that the notion is a structural one rather than depending on the particular objects used to model contexts.

A relevant alternative to Stalnaker's kind of approach, one associated with the dynamic tradition in semantics going back to Heim [1982], proceeds differently. Rather than first supplying a static semantics for the language, and then recovering an associated update system by assuming a simple rule like Stalnaker Update, theorists in the dynamic tradition directly identify the meanings of sentences with CCPs. There is no static semantic middleman on the road to the update system for the language; instead, a (compositional) specification of the update system is itself understood to give the meanings of the sentences of the language.

In choosing between (a) giving a static semantics, and inducing the update system from it along Stalnakerian lines, or (b) giving a dynamic semantics, thereby supplying the language with an update system directly, it is natural to ask: are there phenomena in natural language that cannot be adequately modeled if a static update system is assumed? This question in turn raises to a prior question:

> What general properties are characteristic of a static update system?

This is equivalent to the question: what are the properties whose absence makes for dynamicness? If we can find some general properties necessary and sufficient for staticness, then we can investigate whether natural language supplies counterexamples to these properties; and we can settle whether any given proposed dynamic semantics admits of static reformulation without having to construct a static semantics explicitly.

2 van Benthem staticness

In the literature on this question, there are two notable results. The first is due to van Benthem [1986], and is perhaps the most commonly cited observation on this issue (see, e.g., Groenendijk and Stokhof [1991], van Benthem [1996], von Fintel and Gillies [2007], van Eijck and Visser [2010], Muskens et al. [2011]). It will be convenient to define the notion of a *van Benthem static* update system:

Def 4. An update system $\langle L, B, [\cdot]\rangle$ is VAN BENTHEM STATIC iff there exists a Boolean algebra[3] B_A, $B_A = \langle B, \wedge, \vee, \neg, \top, \bot\rangle$, such that for all $c \in B$ and $s \in L$,

> **Eliminativity.** $c[s] \vee c = c$
>
> **Finite distributivity.** $(c \vee c')[s] = c[s] \vee c'[s]$

Call any such triple $\langle L, B_A, [\cdot]\rangle$ a van Benthem static update system WITH BOOLEAN STRUCTURE.

Then we can state the observation as follows:

Fact 1 (van Benthem 1986)**.** If $\langle L, B_A, [\cdot]\rangle$ is a van Benthem static update system with Boolean structure, where $B_A = \langle B, \wedge, \vee, \neg, \top, \bot\rangle$, then for all $c \in B$ and $s \in L$: $c[s] = c \wedge \top[s]$.[4]

---

[3]A BOOLEAN ALGEBRA is a tuple $\langle B, \wedge, \vee, \neg, \top, \bot\rangle$, where B is a set, $\wedge, \vee$ are binary operations on B, $\neg$ is a unary operation on B, and $\top, \bot \in B$, such that: for any $x, y \in B$: (1) $x \vee (x \wedge y) = x$; (2) $x \wedge (x \vee y) = x$; (3) $x \vee \neg x = \top$; (4) $x \wedge \neg x = \bot$.

[4]See van Benthem [1986, p.86], where the point is made for set algebras in the context of a discussion of intersective adjectives.

We note that some authors, such as van Eijck and Visser [2010], cite van Benthem [1989] for Fact 1. That paper, however, presents a distinct claim:

> If $\langle C, \wedge, \vee, \neg, \top, \bot\rangle$ is a Boolean Algebra which is idempotent and distributive then "the whole information structure can be represented by a set structure of the 'eliminative' kind described earlier" [van Benthem, 1989, p. 38].

It is not entirely clear to us what the statement in quotes means, but from the context it seems that the weakest possible interpretation is as follows: there is a set S and a injective mapping f from C to $\mathcal{P}(S)$ such that for all $c \in C$ and $s \in L$, $f(cs) \subseteq f(c)$. If there is such an f we will say there is an ELIMINATIVE REPRESENTATION of $\langle C, \wedge, \vee, \neg, \top, \bot\rangle$.

The claim is not true. For a counterexample let $C = \{\emptyset, \{a\}\}$, the Boolean operations have their usual interpretation for the powerset algebra of $\{a\}$, and $L = \{s_1, s_2\}$ such that for all $c \in C$, $c[s_1] = c\backslash\{a\}$ and $c[s_2] = c \cup \{a\}$. s_1 and s_2 are trivially distributive and idempotent, but there is no set-theoretic interpretation of this semantics. It is easy to see that the following three inconsistence properties would be needed for there to be an eliminative representation of this semantics: $f(\{a\}) \neq f(\emptyset)$ (since f is injective), $f(\emptyset) \subseteq f(\{a\})$ (since $\{a\}[s_1] = \emptyset$), and

Proof.
$$\begin{aligned} c \wedge \top[s] &= c \wedge (c \vee \neg c)[s] \\ &= c \wedge (c[s] \vee \neg c[s]) && \text{(Finite distributivity)} \\ &= (c \wedge c[s]) \vee (c \wedge \neg c[s]) \\ &= c[s] \vee \emptyset && \text{(Eliminativity)} \\ &= c[s] \end{aligned}$$
□

If our update system is van Benthem static, then the update impact of any sentence s on c can be factored into two steps: first, let the sentence perform its update on the context corresponding to Boolean $\top$; second, take the resulting context and output the Boolean meet of it with c. Now the analogy to a Stalnaker update system should be clear: in a system like this each sentence can be associated with some element in the space of contexts, and the update of every sentence on any c is equivalent to the Boolean conjunction of that element with c. And indeed, we can establish that any van Benthem static update system is isomorphic to some Stalnaker static update system.

Fact 2. If an update system is van Benthem static, it is static.

(We will prove this claim below, as a corollary of a later result.) This supplies us with an illuminating, highly general sufficient condition for staticness.

The question now arises about the converse of Fact 2. If an update system is static, does it follow that it is van Benthem static? The answer is negative. Staticness and van Benthem staticness do not coincide.

Fact 3. There exist update systems which are static, but not van Benthem static.

This follows trivially from the fact that the set of contexts in a static update system needn't form a Boolean algebra. To see this, one need only consider any Stalnaker static update system with finitely many contexts n, such that n does not equal a power of 2.

We can agree that many interesting update systems admit of a natural Boolean structure, and that in many such cases it will be clear that when evaluated with respect to that structure, the system will be van Benthem static. But we can also agree that if an update system fails to be van Benthem static with respect to a

---

$f(\{a\}) \subseteq f(\emptyset)$ (since $\emptyset[s_2] = \{a\}$). (Note that van Benthem [1989] also mentions what he calls monotonicity in the context in which this proof arises: for all $s \in L, c, c' \in C$: $c \leq c'$ only if $c[s] \leq c[s']$. Adding this property to the claim does not help however, as the counterexample here also satisfies monotonicity.)

particular way of equipping it with Boolean structure, nothing yet follows. To conclude a system is not van Benthem static, we must check every possible way of equipping the system with Boolean structure. And even if we do find that a system is not van Benthem static, it does not follow that it is not static. For again, there will be update systems which simply don't have Boolean structure, and in such cases van Benthem staticness does not usefully apply. While it is not unnatural to expect the space of informational contexts to have Boolean structure, who knows what future semantics and pragmatics will reveal? Thus it would be preferable to seek a result which assume as little about the structure of contexts as possible. These observations lead us to ask whether greater generality can be achieved.

3 Veltman staticness

Veltman [1996] answers this question affirmatively. This bring us to our second result. Rather that assuming Boolean structure on the space of contexts, Veltman assumes only that it forms an *information lattice*:

Def 5. A quadruple $\langle V, \top, \wedge, \leq \rangle$ is an INFORMATION LATTICE iff V is a set, $\top \in V$, $\wedge$ is a binary operation on V, and $\leq$ is a partial order on V such that:

$$\top \wedge c = c$$
$$c \wedge c = c$$
$$c \wedge c' = c' \wedge c$$
$$(c \wedge c') \wedge c'' = c \wedge (c' \wedge c'')$$
$$c \leq c' \text{ iff there is some } c'' \text{ such that } c \wedge c'' = c'.$$[5]

Any Boolean algebra determines some information lattice, but not so the converse. (Notably, an information lattice need not include a Boolean $\perp$.) Using these weaker structural assumptions, we define Veltman's notion of staticness as follows:

Def 6. An update system $\langle L, V, [\cdot] \rangle$ is VELTMAN STATIC iff there exists an information lattice, V_I, $V_I = \langle V, \top, \wedge, \leq \rangle$, such that for all $c, c' \in V$ and $s \in L$,

Idempotence. $c[s][s] = c[s]$

Persistence. If $c[s] = c$ and $c \leq c'$ then $c'[s] = c'$

[5]The specification of $\leq$ adds no structure as it is induced by $\wedge$, but we will find the explicit specification convenient below. $c \leq c'$ is meant to be read roughly as "c' is at least as informationally strong as c". Note that $\top$ is being treated as the *minimal* element in this ordering.

Strengthening. $c \leq c[s]$

Monotony. If $c \leq c'$ then $c[s] \leq c'[s]$

Call any such triple $\langle L, V_I, [\cdot]\rangle$ a Veltman static update system WITH INFORMATION STRUCTURE.

Now we observe a result analogous to van Benthem's.

Fact 4 (Veltman 1996)**.** If $\langle L, V_I, [\cdot]\rangle$ is Veltman static update system with information structure, where $V = \langle V, \top, \wedge, \leq\rangle$, then for all $c \in V$ and $s \in L$: $c[s] = c \wedge \top[s]$.

Proof. $c \leq c \wedge \top[s]$

$$\begin{array}{rll}
c[s] & \leq (c \wedge \top[s])[s] & \text{(Monotony)} \\
c[s] & \leq c \wedge \top[s] & \text{(Idempotence, Persistence)}
\end{array}$$

For the other direction:

$$\begin{array}{rll}
\top & \leq c[s] & \\
\top[s] & \leq c[s] & \text{(Idempotence, Monotony)} \\
c \wedge \top[s] & \leq c[s] \wedge c & \\
c \wedge \top[s] & \leq c[s] & \text{(Strengthening)} \\
c \wedge \top[s] & = c[s] &
\end{array}$$

□

As with van Benthem staticness, if our update system is Veltman static, then the update impact of any sentence s on c can be factored into two steps: first, let the sentence perform its update on the context corresponding to $\top$; second, take the resulting context and output the meet of it with c. Thus we have the same analogy to a Stalnaker update system, but against fewer structural assumptions. And indeed, we can establish that any Veltman static update system is isomorphic to some Stalnaker static update system.

Fact 5. If an update system is Veltman static, it is static.

(We will prove this claim below, as a corollary of a later result.) This supplies us with another illuminating sufficient condition for staticness, and one more general than van Benthem staticness.[6] (Admittedly it is comparatively more complex, involving as it does the joint satisfaction of four properties rather than two.)

[6] Given Fact 1 it is easy to verify that any van Benthem static system determines an information lattice wherein idempotence, persistence, strengthening, and monotony hold, and hence van Benthem staticness implies Veltman staticness. The converse does not hold, as an information lattice need not admit of Boolean structure.

Again, the natural next question concerns the converse of Fact 5. If an update system is static, is it Veltman static? The answer is again negative: staticness and Veltman staticness do not coincide.

Fact 6. There exist update systems which are static, but not Veltman static.

This follows straightforwardly from the fact that the set of contexts in a static update system needn't form an information lattice. To see this, one need only note that a Stalnaker static update system does not have to contain an element playing the $\top$-role.

Example. Consider the static update system $\langle C, L, [\cdot]\rangle$ such that $C = \{\{a\}, \{b\}, \emptyset\}$, $L = \{a, b\}$, and for all $c \in C, s \in L : c[s] = c \cap \{s\}$. Obviously, for every $c \in C$ either $c[a] = c$ or $c[b] = c$. Now suppose for contradiction that this system is Veltman static. Then there exists $\top \in C$; hence either $\top[a] = \top$ or $\top[b] = \top$. Suppose $\top[a] = \top$. Then, by Fact 4, for all $c \in C$, $c[a] = c \wedge \top[a] = c \wedge \top = c$. But not so, since $\{b\}[a] \neq b$. So $\top[a] \neq \top$. By symmetry $\top[b] \neq \top$. Hence $\top[a] \neq \top$ and $\top[b] \neq \top$. Contradiction.

$\square$

Now the general thrust of our comments on van Benthem staticness apply *mutatis mutandis* to Veltman staticness. We can agree that many interesting update systems admit of a natural information lattice structure, and that in many such cases it will be clear that when evaluated with respect to that structure, the system will be Veltman static. But we can also agree that if an update system fails to be Veltman static with respect to a particular way of equipping it with information lattice structure, nothing yet follows. To conclude a system is not Veltman static, we must check every possible way of equipping the system with information lattice structure. And even if we do find that a system is not Veltman static, it does not follow that it is not static. For again, there will be update systems which simply don't information lattice structure, and in such cases Veltman staticness does not usefully apply.

It should be uncontroversial that the ideal would be to be able state conditions which are both sufficient *and necessary* for staticness (equivalently, dynamicness). If we have only sufficient conditions for staticness, it follows we lack sufficient conditions for dynamicness; and that in turn implies we have yet to demarcate the static-dynamic boundary. It is natural to ask again, then, whether greater generality can be achieved.

4 A representation theorem for staticness

Fortunately, the desired level of generality can be achieved. This takes us to the main result of the paper, a representation theorem for staticness. We show an update system is static if and only if it has the properties of *idempotence* and *commutativity*.

Fact 7 (Static representation theorem)**.** An update system $\langle L, C, [\cdot]\rangle$ is static iff for all $s \in L$ and $c \in C$,

Idempotence. $c[s][s] = c[s]$
Commutativity. $c[s][s'] = c[s'][s]$

We begin with the right-left direction.

Fact 7.1 If an update system is idempotent and commutative, then it is static.

Proof. Let $\langle L, C, [\cdot]\rangle$ be an idempotent and commutative update system. To show that this system is static, it suffices to show that there exists a static semantics $\langle L, W, [\![\cdot]\!]\rangle$ an injective function f, $f : C \to \mathcal{P}(W)$, such that $f(c[s]) = f(c) \cap [\![s]\!]$, for all $s \in L$ and $c \in C$.

Before defining f, we first define a relation $\leq$ as follows:

Def 7. For any update system $U = \langle L, C, [\cdot]\rangle$, and $c, c' \in C$, by definition $c \leq_U c'$ iff there exist $s_1 \ldots s_n \in L$ such that $c[s_1]....[s_n] = c'$, or $c = c'$. (We will just write $\leq$ if the update system being discussed is clear from context.)

We will find the following abbreviation useful: since $[\cdot]$ is commutative, we can speak of the update of a set of sentences on a context irrespective of their sequential order:

Def 8. If S is a finite set of sentences $s_1....s_n$ from L, $c[S] =_{\text{DEF}} c[s_1]....[s_n]$.

We pause to observe $\leq$ is transitive, reflexive and anti-symmetric. Reflexivity is trivial. Transitivity: suppose $c_1 \leq c_2$ and $c_2 \leq c_3$. Then for some S, S', $c_1[S] = c_2$ and $c_1[S'] = c_3$; hence $c_1[S][S'] = c_3$, so $c_1 \leq c_3$. Anti-symmetry: suppose $c_1 \leq c_2$ and $c_2 \leq c_1$. Then for some S, S', $c_1[S] = c_2$ and $c_2[S'] = c_1$, and hence $c_1[S][S'] = c_1$. By commutativity it follows that $c_1[S'][S] = c_1$, and hence $c_1[S'][S][S] = c_1[S]$. By idempotence $c_1[S'][S][S] = c_1[S'][S]$, so substituting, $c_1[S'][S] = c_1[S]$; substituting again, $c_1 = c_2$.

Let $f : C \to \mathcal{P}(C)$ as follows: $f(c) = \{c' \in C : c \leq c'\}$. We need to show that f is injective, i.e., if $f(c_1) = f(c_2)$ then $c_1 = c_2$, for all $c_1, c_2 \in C$. Suppose $f(c_1) = f(c_2)$.

Now $f(c_1) = \{c' \in C : c_1 \leq c'\}$, hence by reflexivity $c_1 \in f(c_1)$. Hence $c_1 \in f(c_2)$; hence $c_1 \in \{c' \in C : c_2 \leq c'\}$ and therefore $c_2 \leq c_1$. By parity, $c_2 \in f(c_1)$, and $c_1 \leq c_2$. By anti-symmetry, $c_1 = c_2$.

Now define a static interpretation function $[\![\cdot]\!] : L \to \mathcal{P}(C)$ to be the minimum function such that $[\![s]\!] = \{c \in C : c[s] = c\}$. (Thus $[\![\cdot]\!]$ takes s to its fixed points on the update function $[\cdot]$.)

The preceding defines (i) a static semantics $\langle L, C, [\![\cdot]\!]\rangle$ given an arbitrary commutative idempotent update system $\langle L, C, [\cdot]\rangle$, and (ii) a injective function f from $C \to \mathcal{P}(C)$. It remains to show that for all $c \in C$ and $s \in L$, $f(c[s]) = f(c) \cap [\![s]\!]$.

First we show that if $c_1 \in f(c[s])$, then $c_1 \in f(c) \cap [\![s]\!]$. Suppose $c_1 \in f(c[s])$. (i) Then $c_1 \in \{c' \in C : c[s] \leq c'\}$. So $c[s] \leq c_1$. By definition $c \leq c[s]$. So $c \leq c[s] \leq c_1$. Hence by transitivity $c \leq c_1$, hence $c_1 \in f(c)$. (ii) Now since $c[s] \leq c_1$, there exists some S such that $c[s][S] = c_1$. So $c[s][S][s] = c_1[s]$. By commutativity, $c[s][S][s] = c[S][s][s]$, which by idempotence equals $c[S][s]$, which by commutivity equals $c[s][S]$. So $c[s][S][s] = c[s][S]$. Here we substitute c_1 for $c[s][S]$, and we have $c_1[s] = c_1$. From this it follows that $c_1 \in [\![s]\!]$, since the latter just is $\{c \in C : c[s] = c\}$. So from (i) and (ii) we have $c_1 \in f(c) \cap [\![s]\!]$, the desired result.

Now let us show that if $c_1 \in f(c) \cap [\![s]\!]$, then $c_1 \in f(c[s])$. This is equivalent to showing that if $c_1[s] = c_1$ and $c \leq c_1$, then $c[s] \leq c_1$. Suppose $c \leq c_1$. Then there is some S such that $c[S] = c_1$. Suppose also $c_1[s] = c_1$. Then we have $c[S] = c_1[s] = c_1$. Therefore $c[S][s] = c_1$. By commutativity $c[s][S] = c_1$. And that means $c[s] \leq c_1$; and therefore $c_1 \in f(c[s])$.

□

The left-right direction completes the proof:

Fact 7.2 If an update system is static, then it is commutative and idempotent.

Proof. Any static system is clearly idempotent and commutative, since intersection is idempotent and commutative. □

This theorem effectively supplies proofs of Fact 2 (If an update system is van Benthem static, it is static) and Fact 5 (If an update system is Veltman static, it is static). We need only observe that any system which is van Benthem or Veltman static is also idempotent and commutative. Now since any van Benthem static system is also Veltman static, it suffices to show that any Veltman static system is idempotent and commutative; and since any Veltman static system is idempotent by definition, it suffices to observe that if an update system is Veltman static, it is commutative. This point is easy to see: given Fact 4, for any Veltman static system $\langle L, V, [\cdot]\rangle$, there is an information lattice V_I, $V_I = \langle V, \top, \wedge, \leq\rangle$ such that for all $c \in V$ and $s \in L$. $c[s] = c \wedge \top[s]$. Since $\wedge$ is a commutative operator, for any $s, s' \in L$ and $c \in V$, $c[s][s'] = c[s'][s]$.

The theorem also makes it easy to say in virtue of what various existing dynamic semantic systems are essentially non-static. We can observe, for instance, that file change semantics (Heim [1982, 1983a]), dynamic predicate logic (Groendijk and Stokhof [1991]), and update semantics (Veltman [1996]) are all dynamic fundamentally because they permit violations of commutativity.[7]

5 Idempotence and Commutativity

We set out to answer the question what general properties are characteristic of a static update system. We have our answer: the static update systems are the commutative, idempotent update systems. The obvious question now is whether natural languages supply counterexamples to these properties. Both of these are very intuitive properties of natural language: Asserting the same sentence twice, is simply being redundant. The only possible difference from a single assertion would seem to be a matter of pragmatics, not semantics (e.g. repetition might be used for emphasis, viz. 'Yes we can! Yes we can!'). So idempotence is an extremely natural semantic assumption. Commutativity is also obvious: semantically speaking the same information is conveyed by an assertion of two sentences in either order; the choice of order would seem to be just a stylistic one. Nonetheless, as we shall see, superfical counterexamples to both properties abound; thus the real question is whether such putative counterexamples can be explained, or explained away, along static lines.

Our discussion here is not intended as a thorough investigation of whether a dynamic or static semantics is, overall, the best theory. The resolution of that will depend on detailed empirical analysis of particular systems, and cannot be determined by a merely superficial look at potential linguistic examples of non-idempotent or non-commutative constructions. Nonetheless, the static representation theorem allows us to see the commonalities amongst all dynamic semantics. If natural language

[7]Examples can be found in the appendix.

provides clear examples of failures of idempotence or commutativity that would provide, in itself, a prima facie reason for thinking that the best semantics will be a dynamic one. If, on the other hand, idempotence and commutativity seem to be properties of natural language, that is a prima-facie consideration in favor of a static semantics. The purpose of general results like the static representation theorem is they allow us to draw empirical consequences from very high-level theoretical choices. Thus, they allow us to make a prima-facie assessment of what framework we should choose for our semantic theorizing. The discussion of this section is intended at this general level; it is not intended as a detailed empirical comparison of extant static or dynamic semantics.[8]

5.1 Idempotence

Let us begin with three superficial counterexamples to idempotence. The first example involves flatfooted use of demonstratives. Compare the state of the discourse at the end of (1-a) to the state of the discourse after (1-b):

(1) a. (a) This [*pointing to the lamp*] is old.
b. This [*pointing to the table*] is old.

Plainly, (1-b) adds more information to the common ground. Second example:

(2) You and a friend are trying to count the prime numbers between one and twenty. You have counted five so far. You say:
a. I thought of another prime.
There is a pause, then you say again:
b. I thought of another prime.

Intuitively, the second utterance, (2-b), seems to add information to the common ground not previously there. Third example:

(3) You are at a taco stand and the man behind the counter is making your tacos and has just put some guacamole on them. You say:

[8] In this respect the discussion parallels the issues Chomsky raised about context-free grammars (references).

a. Please add some more guacamole.
He puts some additional guacamole and you say again:
b. Please add some more guacamole.

Again the second utterance seems to add something more, rather than just being redundant.

There are various ways one might attempt to accommodate these examples along static lines. A familiar kind of reply to the first example, building on Kaplan [1977/1989], maintains simply that the demonstrative *this* is context-sensitive, in the sense that its contribution to the determination of the proposition expressed, and added to the common ground, is a function of the context. Because of the way the context changes after the utterance of the first *this*, the contribution of the second *this* to the proposition the second sentence expresses differs from that of the first. Thus, each sentence ends up expressing a different proposition; hence their cumulative context-change effect is different from what we would have had if a single proposition were expressed twice-over.

It is not obvious whether this sort of reply can be extended to the examples (2) and (3). With respect to (2), it is less clear that any element is context-sensitive than it was in (1). However it might plausibly be claimed that *another* includes an indexical reference of some sort.[9] With (3) it is not as clear that there are any context sensitive terms. Nonetheless, there is another reason why this is not a good counterexample to idempotence. It seems like the context changes between the first utterance and the second in this case. But if there is an independent shift in the context between utterances this is not a true instance of a failure of idempotence.[10] **Daniel's comment**: A subtle point arises here as staticness entails that once one update is made it will be redundant from then on. So in a sense this *is* a good counterexample to staticness, but not via a failure in idempotence... Rather what you have is a potential counterexample to a formal property entailed by idempotence and commutativity together. I guess we should maybe address this? It'll be a pain though.

[9]This would fall into line with Kripke's [2009] observations about presupposition triggers, such as *too*.

[10]We could try to say the same thing, about (3), though here it is less clear what the shift to the context is, besides the first assertion having been made.

5.2 Commutativity

Dynamic semantics is not generally motivated by alleged failures in idempotence: in fact, all of the prominent dynamic systems satisfy idempotence. Most empirical motivations for dynamic semantics are based, directly or indirectly, on failures of commutativity. In this section, we discuss the three main linguistic phenomenon that have motivated non-commutativity: presupposition, the treatment of variables, and epistemic modals.

5.2.1 Presupposition

It is a basic observation in the presupposition literature that the order in which two sentences are uttered can determine whether or not they (as a pair) lead to a presupposition failure. For instance, in any context the following two sentences can be asserted in this order without leading to a presupposition failure:

(4) I used to smoke. Last month I stopped.

However if we reverse the order, it is often argued that we may (in some contexts) have a presupposition failure due to the use of *stop* in the first sentence.

(5) Last month I stopped. I used to smoke.

Similar phenomenon occur with pronouns such as *he*.

(6) a. A man walked in. He was tired.
b. He was tired. A man walked in.

We take the order sensitivity of presuppositions in these examples for granted. The question is whether this data motivates a non-commutative semantics. Heim [1982, 1983b] proposed a non-commutative update system as a treatment of these basic examples, as well as more complex cases of presupposition projection. This basic approach has been extremely influential.

However, there are serious alternatives to a non-commutative semantics to handle the data above. Stalnaker [1973, 1974] forcefully argued that many order effects involving presupposition should be explained at the pragmatic level rather than the semantic one. Schlenker [2006, 2008] scales up a pragmatic story into a serious empirical rival to the most sophisticated dynamic accounts and he claims empirical and theoretical advantages for it. Schlenker [2008, 2009], Rothschild [2008, 2011] go further and argue that in many instances presupposition projection is commutative, something that cannot naturally be accounted for on standard dynamic frameworks. Thus, it remains very much an open question whether the order sensitivity of presupposition provides a compelling motivation for dynamic semantics.

5.2.2 Reusing variables

A more subtle argument for non-commutativity involves the treatment of variables in dynamic semantics. This argument requires some presentation of the basic syntax of Heim's file change semantics. One of the many innovations in Heim's systems, paralleled in Kamp [1981], is the uniform treatment of indefinite descriptions, definite descriptions and pronouns as variables. Heim's treatment of variables leads to the only failures of commutativity in her system besides those related to presupposition projection.[11] For instance, in her system (which is presented in the appendix) Fx and $\neg Gx$ are not commutative with respect to order of update.

We might wonder whether this class of failures of commutativity in update is motivated by empirical data. If we accept the syntactic move of treating all descriptions as variables, then a sort of motivation can be found. A sentence such as Fx might be the logical form of the English sentence *A man came in*, while $\neg Gx$ might be the logical form of the sentence *He didn't buy a drink*. Saying a *A man came in...He didn't buy a drink* leads to a recognizable update. Given our syntax, $\neg Gx$ also can also be the logical form of the sentence *A man didn't buy a drink*. The normal reading of this (and the one that Heim's system captures) is the reading on which it is equivalent to *No one bought a drink*. On the assumption that $\neg Gx$ is the common logical form of the two superficially distinct sentences, we can get a kind of commutativity failure. To see this, just note that (for a fixed domain) accepting that *A man came in...He didn't buy a drink* is possible in some contexts where accepting *No man bought a drink* is not possible at all (no matter what it is followed by), and hence there is failure of commutativity.

We think it is clear that this kind of phenomenon does not provide a compelling

[11] A similar thing might be said about DRT but the matter is somewhat complicated by Kamp's syntactic restrictions on the reuse of variables which makes certain potential examples of non-commutativity inexpressible.

direct empirical argument for non-commutativity. Rather what we have is a non-commutativity in Heim's semantics that can at least be related to some aspects of the semantics of natural language. The data though does not cry out for a non-commutative treatment. We are not forced, for instance, to think that definite descriptions are syntactically identical to indefinite descriptions. For the case to be made for this kind of non-commutativity based on these type of examples we would need much support for the syntactic and semantic assumptions underlying the treatment of variables in Heim's semantics. We will not enter into this complex theoretical debate: we are here limiting our attention to more direct empirical motivations for the framework.

5.2.3 Epistemic modals

Epistemic modals provide the sole case—to our knowledge—in which examples of non-commutativity have explicitly been cited as motivations for a dynamic semantics. Veltman [1996] motivated his dynamic semantics for epistemic modals, which is non-commutative, by reference to pairs like this:[12]

(7) a. Billy might be at the door... it isn't Billy at the door.
    b. It isn't Billy at the door... Billy might be at the door.

The empirical claim is that (7-a) is a felicitous bit of discourse, while (7-b) is not. Veltman's dynamic semantics (as outlined in the appendix) captures this easily as (7-a), $c[\Diamond P][\neg P]$, defines an update that can be successful, whereas (7-b), $c[\neg P][\Diamond P]$ defines an update that always crashes (i.e. takes one to the empty context). Thus, the dynamic account predicts the supposed difference between these discourses.

The difference between these discourse, however, might easily be predicted by a wide class of theories. For consider this pair:

(8) a. I think it's possible that Billy is at the door... Billy isn't at the door.
    b. Billy isn't at the door... I think it's possible Billy is at the door.

Few would argue that anything in (8), requires a dynamic treatment: we have an attitude ascription and then a factual statement. However, the order in which they

[12] Pairs like this are cited approvingly, as motivating a dynamic semantics, by Groenendijk et al. [1995], Veltman [1996], Beaver [2001], Gillies [2001], and von Fintel and Gillies [2007], among others.

are presented leads to dramatic differences in how coherent the piece of discourse is. (8-a) seems to reflect standard information growth, while (8-b) indicates I kind of retraction (and, thus, is a bit odd without something marking that like 'Wait, I was wrong!' or 'No,'). It would seem a real failure of theoretical imagination to think that a static picture of communication could not explain the difference between information growth and retraction, which seems the only significant difference in either (7-a) and (7-b) or (8-a) and (8-b).

6 Dynamic versus context-senstitive static

We noted above that certain prima-facie counterexamples to idempotence, might be instead explained by an appeal to context-sensitivity. Such a move may also be available in the face of prima-facie failures of commutativity. For example it is possible for a theorist to argue that a context-sensitivity in the epistemic modal *might* accounts for any difference between (7-a) and (7-b).[13]

Such an appeal to context-dependency raises the important foundational question of how we treat the context-sensitivity of linguistics expressions in an update system. One thing we could say is that at the relevant level of abstraction, the two sentences in, for example, (1) are not actually of the same sentence type. Simple demonstratives, one might try saying, are analogous to variables in that at the relevant level of syntactic analysis, they bear indices. Moreover, in this case, the indices of the respective demonstratives are distinct, with context associating each of them with distinct referents. Thus we should write instead:

(1) a. This_1 is old.
    b. This_1 is old. This_2 is old.

On this view, (8-b) does not contain the same sentence twice over; hence it is not a counterexample to idempotence. This blocks the counterexample by an appeal to a covert syntactic features which are context-sensitive.

However, associating different syntactic structures is not the only, or even the most natural, way of treating all instances of context sensitivity. So we should look more carefully at other options.

[13] This is one way of reading the semantics presented in Yalcin [2007].

We begin by discussion the standard Kaplanian framework for treating context-sensitivity. On this view: the interpretation function is relativized to contexts, where contexts are now understood to be (not as common grounds but) as *discourse locations*. Contexts in this sense determine an agent, time, position, and possible world; they correspond roughly to the centered worlds of Lewis [1979]. Assume what we can call a *Kaplan semantics*:

Def 9. A quadruple $\langle K, W, L, [\![\cdot]\!]\rangle$ is KAPLAN SEMANTICS iff $[\![\cdot]\!]$ is a function from L into $K \times W$, where K is the set of discourse locations and W the space of possible worlds.

On this approach, sentences s only determine a proposition expressed relative to a discourse location k; we could write this proposition $[\![s]\!]^k$. Since, by usual static lights, the context-change potential of a sentence is a function of the proposition the sentence expresses, this means that generally speaking, sentences only have context-change potentials relative to a choice of location. Maintaining the Stalnakerian idea of representing the common ground of a conversation as a set of possible worlds, we could write the corresponding update rule as follows:

> **Location-Sensitive Stalnaker Update.** For all common grounds c, discourse locations k, and sentences s: $c[s_k] =_{\text{DEF}} c \cap [\![s]\!]^k$.

(This is in fact closer to the picture of Stalnaker [1978] than the rule of Stalnaker Update in section 1.) Observe that this update rule is not a function from sentences to context-change potentials, but rather a function from sentences *indexed to locations*—equivalently, sentence-location pairs—to context-change potentials. This is one way of seeing that this approach effectively rejects an idea we assumed at the outset, namely that any given natural language determines some update system or other. One cannot generally associate sentences with context-change potentials in abstraction from features of the specific discourse location. As a result, we cannot really raise the question of whether the system is static or dynamic, in the sense we have defined. From one perspective, what this shows is that our definition of staticness is of highly limited application; for while Kaplan's influential semantics certainly *looks* static, it emerges according to our definition that it does not induce a static update system—or indeed, *any* update system.

Before we draw this conclusion, consider now a second way of glossing "context-sensitivity." The second way accords with our use throughout the paper: a context correspond to a common ground, to the sort of thing which is updated by a context-change potential. Suppose that sentences express propositions only relative to a context in this sense. On the hypothesis that propositions and contexts can be modeled as sets of possible worlds, the idea would that that any given natural language can be associated with a *context-sensitive semantics*:

Def 10. A triple $\langle W, L, [\![\cdot]\!]\rangle$ is a CONTEXT-SENSITIVE SEMANTICS iff $[\![\cdot]\!]$ is a function from L to the set of functions from $\mathcal{P}(W)$ to $\mathcal{P}(W)$.

Given such a semantics, we write the proposition denoted by $s \in L$ relative to common ground $c \in \mathcal{P}(W)$ as $[\![s]\!]^c$. Then we can state a rule associating sentences with context-change potentials in the spirit of Stalnaker Update as follows:

> **Context-Sensitive Stalnaker Update.** For all common grounds c and sentences s: $c[s] =_{\text{DEF}} c \cap [\![s]\!]^c$.

The intuitive idea here is that at least some sentences will determine a proposition expressed only relative to the prior presuppositions of the discourse; or again, the idea is that as far as specifying the context-change potential of a sentence goes, the only item of information which might be needed about the discourse location is the common ground of the discourse.

This picture, unlike the Kaplanian picture recently described, can be seen to straightforwardly determine a class of update systems. We call this the class of *context-sensitive* update systems:

Def 11. An update system $\langle L, C, [\cdot]\rangle$ is CONTEXT-SENSITIVE if and only if there exists a context-sensitive semantics $\langle L, W, [\![\cdot]\!]\rangle$ and a one-to-one function f from C to $\mathcal{P}(W)$ such that for all $c \in C$ and $s \in L$, $f(c) \cap [\![s]\!]^{f(c)} = f(c[S])$

It should be clear that context-sensitive semantics is a generalization of the notion of static semantics. Static semantics are the special case where the semantic denotations are not affected by the context.

We can now ask what formal properties characterize context sensitive update systems.

There is, it turns out, a very simple answer to this question. We give a formal characterization of context sensitive update system, in terms of the relation $\leq$ defined on page 9.

Fact 8. For any update system $U = \langle L, C, [\cdot]\rangle$, U is CONTEXT-SENSITIVE iff $\leq$ is a partial order.

Proof. We first show if U is context-sensitive then $\leq$ is a partial order. Reflexivity and transitivity are immediate consequences of the definition of $\leq$. For anti-symmetry simply note that for c, d in C' $c \leq d$ only if $c \supseteq d$. It follows that if $c \leq d$ and $d \leq c$, $c = d$.

We now show that if $\leq$ is a partial order then U is context-sensitive. Let $f : C \to \mathcal{P}(C)$ be such that $f(c) : \{c' : c \leq c'\}$. Note that f is injective, as we showed in our proof of Fact 7 using only the fact that $\leq$ is a partial order. Now consider the context-sensitive semantics $\langle L, C, [\![\cdot]\!]\rangle$ where $[\![s]\!]$ is the minimal function such that for all $d \in f(C)$ and $s \in L$, $[\![s]\!]^d = f(f^{-1}(d)[s])$. It remains to show that for all $c \in C$ and $s \in L$, $f(c) \cap [\![s]\!]^{f(c)} = f(c[S])$. First, we show that $[\![s]\!]^{f(c)} \subseteq f(c)$. Note first that $[\![s]\!]^f(c) = f(f^{-1}(f(c))[s]) = f(c[s])$ and $c \leq c[s]$. Suppose $c' \in f(c[s])$, then, by definition, $c[s] \leq c'$. So, by transitivity of $\leq$, $c \leq c'$,and thus $c' \in f(c)$. So $[\![s]\!]^{f(c)} \subseteq f(c)$. It follows immediately that: $f(c) \cap [\![s]\!]^f(c) = [\![s]\!]^f(c) = f(c[s])$.

□

What Fact 8 shows is that as long as information states can be given a partial order with respect to updates then an update system is context-sensitive. For this reason the class of semantics that are context-sensitive is extremely broad, including all static semantics and all dynamic semantics in the literature. Thus, interestingly it turns out that the kind of data motivate the dynamic turn, in fact motivates only a limited form of dynamicness, one that is still isomorphic to a special kind of context-sensitive static semantics.

7 Conclusion

Our purpose has been to take a direct look at the basic framework behind dynamic semantics to see what empirical properties of a language would necessitate a rejection of static semantics. We saw that the two necessary and sufficient properties for an update system to have in order to be static are idempotence and commutativity. This formal characterization of staticness (and hence of it dynamicness) allows us to go beyond the typical rhetoric used to describe dynamic semantics as treating meanings like ‘computer programs’.

This formal result suggests that the most basic way to see whether we should adopt a dynamic semantics is to see whether natural language has failures in idempotence and commutativity. We did not find compelling examples of either. This is not to say that dynamic semantics cannot be motivated: but that a direct empirical case from the formal properties inherent to the system is not easy to make.

Our discussion of potential counterexamples to idempotence and commutativity led us to consider how we could incorporate an adequate account of context-sensitivity within the framework of update systems. We defined the notion of a context-sensitive update system, which proved to be very general, incorporating all current dynamic systems.

We note here something little discussed in the literature. While the feature of dynamic semantics often emphasized is its dynamicness, there are several other features in Heim and Kamp's systems (as well as later extensions of them) which are independent of dynamicness but lead to much of the power of the systems. These include: unselective quantification, the uniform treatment of pronounds/indefinites as variables (both related ideas from Lewis [1975]), and notion of discourse referent/enrichment of notion of context (building on Karttunen's classic paper, but very much original to Heim/Kamp also). These are all particular aspects of the syntax/semantics used that do not fundamentally require a dynamic system to be implemented. To our knowledge, there have not been systematic static semantics that attempt to incorporate these features of dynamic semantics, but we think exploring such systems might prove worthwhile.

One more note on a direction for future research: here we really defined two basic classes of update systems: the static ones and the context-sensitive ones (and by extension the complements of these classes). The standard dynamic semantics on the market are context-sensitive, but not static. It might be interesting to find intermediate classes of update systems between these two. This could lead to a more refined 'hierarchy' of dynamicness, and allow for a more precise characterization of how far current dynamic systems take us from staticness.

Appendix: Two dynamic update systems

Here we present the basic syntax and semantics for Heim's [1982, 1983a] File Change Semantics (FCS) and Veltman's [1996] semantics for epistemic modals, as well as highlighting instances of non-commutativity in both systems. Our presentation follows the original work, but states them explicitly as *update systems*, as defined in this paper. We refer the readers to Groendijk and Stokhof [1991] for presentation of dynamic predicate logic (DPL), another prominent dynamic semantics.[14]

Heim's File Change Semantics

Here we give a simplified version of Heim's semantics, covering no more than the minimum material needed to make the points above (i.e. atomic formulas and negation).[15]

Def 12. An INTERPRETED FILE CHANGE LANGUAGE is a tuple, $\langle V, P, W, O, I\rangle$, where V is a set of variables, P is a set of n-place predicates, W is a set of worlds, O is a set of objects, and I is an interpretation function which maps an n-place predicate in P and a world in W to a set of n-tuples of elements of O.

If F is an n-place predicate in P, and $w \in W$,we wrote $I(F)(w)$ as $[\![F]\!]^{I,w}$.

[14] Groenendijk and Stokhof present their dynamic predicate logic as the first *compositional* dynamic semantics. They write that no previous semantics in this vein (including Heim's FCS) "makes compositionally its starting point", and then they qualify this with the parenthetic remark that"it seems that Heim [1982, Ch.3], does attach some value to compositionality." This seems ungenerous to us, as Heim's semantic system is clearly and straightforwardly compositional. One would think would think given the importance they accord to compositionality they would explain what lack of compositionality they found in Heim's work. In fact, Groenendijk and Stokhof do not discuss Heim's semantics at all, saying that they "feel justified in restricting comparison to just [Kamp's DRT]" since it is the most "formally explicit" theory in this area. As any careful reader of Heim (1982, Ch 3, 1983a) is aware, the basic semantics there is precise or explicit. The suggestion that Heim's work need not be discussed in this context due to any lack of formal rigor is bizarre. The truth is quite simple: Heim gave a compositional dynamic semantics in her 1982 dissertation (though the term *dynamic semantics* was not yet in currency) and, seven years later, Groenendijk and Stokhof published a different compositional, dynamic semantics. The interest and importance of their work is clear: it does not, however, rest in its claim to being the first compositional dynamic semantics. (We emphasize this point since some later commentators have followed Groenendijk and Stokhof in suggesting Heim's account is somehow not overtly compositional: Geurts and Beaver [2011] label Heim's semantics as "intended to be compositional" with no further explanation, Dekker [1996], while noting that Heim anticipates Groenendijk and Stokhof, describes her as only doing so "implicitly", saying that the compositional presentation is "wrapped up in the representational metaphor of changing files and file cards"—as if presenting a helpful metaphor obscures the technical account. Jäger [1996] makes some interesting, and accurate, comments about how Heim's semantics lacks certain forms of compositionality, but these relate to her felicity conditions on definiteness and the particular level of syntactic representation she works at, and so seem orthogonal to any concerns Groenendijk and Stokhof might have had.)

[15] A more thorough description and discussion of Heim's system is in Yalcin [forthcoming].

Def 13. A VARIABLE ASSIGNMENT is a function from V to O. Let F be the set of all assignment functions. An ASSIGNMENT WORLD is a pair $\langle f, w\rangle$ where $f \in F$ and $w \in W$. SATISFACTION SET is a set of assignment worlds. A DOMAIN is a subset of V. A FILE is a pair $\langle a, d\rangle$ where a is a set of assignment worlds, and d is a domain. For any file $c = \langle a, d\rangle$ we write $c^{\text{sat}} = a$ and $c^{\text{dom}} = d$

We can now define a Heimian update system as follows:

Def 14. For any interpreted file change language, $\langle V, P, W, O, I\rangle$, its HEIMIAN UPDATE SYSTEM is the triplet $\langle L, C, [\cdot]\rangle$ where L is set of all sentenced formed out of V and P and negation the negation sign $\neg$ using the usual syntactic rules of predicate logic, C is the set of all files associated with V, O, W, and $[\cdot]$ is recursively defined as follows for arbitrary $c \in C$:

- for $F(x_1 \ldots x_n) \in L$ and $c[F(x_1 \ldots x_n)] = \langle\{(f, w) \in c^{\text{sat}} : \langle f(x_1) \ldots f(x_n)\rangle \in [\![F]\!]^{I,w}, c^{\text{dom}} \cup \{x_1 \ldots x_n\}\rangle$
- if $\alpha \in L$ then $c(\neg\alpha) = \langle(f, w) \in c^{\text{sat}} : \forall f' \sim_{c^{\text{dom}}} f, (f', w) \notin c[\alpha]^{\text{sat}}, c^{\text{dom}}\rangle$, where $\forall f' \sim_d f$ means: for all assignment functions f' such that f' differs from f only what objects it assigns to variables in d.

It is easy to see that this language is idempotent but not commutative. An instance of a failure of commutativity is with Fx and $\neg Gx$.

Veltman's Semantics for Epistemic Modals

Def 15. An INTERPRETED VELTMAN LANGUAGE is a triple $\langle S, W, I\rangle$ where S is a set of propositional atomic sentences, W is a set of worlds and I is a function from sentences and worlds to truth values.

For $s \in S$, we let $[\![s]\!]^I = \{w : I(s)(w) = \text{true}\}$. We now define a Veltman update system as follows:

Def 16. Given an interpreted Veltman language $\langle S, W, I\rangle$ its VELTMAN UPDATE SYSTEM is the triplet $\langle L, C, [\cdot]\rangle$ where L is the set of all sentences formed out of S, $\neg$ and $\Diamond$ in the usual way, $C = \mathcal{P}(W)$ and $[\cdot]$ is defined recursively as follows for arbitrary $c \in C$:

- if $s \in L$ is in S then $c[s] = c \cap [\![S]\!]^I$

- for $\alpha \in L$, $c[\neg\alpha] = c \backslash c[\alpha]$
- for $\alpha \in L$, $c[\Diamond\alpha] = c$ if $c[\alpha] \neq \emptyset$ otherwise $c[\Diamond\alpha] = \emptyset$

It is easy to see that this system is non commutative. While $c[\neg S][\Diamond S] = \emptyset$ for all c, $c[\Diamond S][\neg S]$ does not.

References

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

Paul Dekker. The values of variables in dynamic semantics. *Linguistics and Philosophy*, 19:211–257, 1996.

Bart Geurts and David I. Beaver. Discourse representation theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2011 edition, 2011.

Anthony Gillies. A new solution to Moore's paradox. *Philosophical Studies*, 105: 237–250, 2001.

J. Groendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14:39–100, 1991.

Jeroen Groenendijk and Martin Stokhof. Two theories of dynamic semantics. *Lectures Notes in Computer Science*, 478:55–64, 1991.

Jeroen Groenendijk, Martin Stokhof, and Frank Veltman. Coreference and modality. In S. Lappin, editor, *Handbook of Contemporary Semantic Theory*. Blackwell, 1995.

Irene Heim. *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, University of Massachusetts, Amherst, 1982.

Irene Heim. File change semantics and the familiarity theory of definiteness. In Christoph Schwarze Rainer Bäuerle and Arnim von Stechow, editors, *Meaning, Use, and Interpretation of Language*. Walter de Gruyter, 1983a.

Irene Heim. On the projection problem for presuppositions. In D. Flickinger and M. Wescoat, editors, *WCCFL 2: Second Annual West Coast Conference on Formal Linguistics*, pages 114–125. Blackwell, 1983b.

Gerhard Jäger. *Topics in Dynamic Semantics.* PhD thesis, Humbolt University Berlin, 1996.

Hans Kamp. A theory of truth and semantic representation. In J. Groenendijk et al., editor, *Formal Methods in the Study of Language*, pages 277–322. Mathematish Centrum, 1981.

David Kaplan. Demonstratives. In J. Almog, J. Perry, and H. Wettstein, editors, *Themes from Kaplan.* Oxford University Press, 1977/1989.

Saul A. Kripke. Presupposition and anaphora: Remarks on the formulation of the projection problem. *Linguistic Inquiry*, 40:367–386, 2009.

David Lewis. Adverbs of quantification. In Edward L. Keenan, editor, *Formal Semantics of Natural Language*. Cambridge University Press, 1975.

David Lewis. Attitudes de dicto and de se. *The Philosophical Review*, 88(4):513–543, 1979.

Reinhard Muskens, Johan van Benthem, and Albert Visser. Dynamics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 607–670. Elsevier, 2nd edition, 2011.

Daniel Rothschild. Presupposition projection and logical equivalence. *Philosophical Perspectives*, 22:473–497, 2008.

Daniel Rothschild. Explaining presupposition projection with dynamic semantics. *Semantics and Pragmatics*, 4(3):1–43, 2011.

Philippe Schlenker. Anti-dynamics: Presupposition projection without dynamic semantics. *Journal of Logic, Language, and Information*, 16:325–56, 2006.

Philippe Schlenker. Be articulate: A pragmatic theory of presupposition projection. *Theoretical Linguistics*, 34:157–212, 2008.

Philippe Schlenker. Local contexts. *Semantics and Pragmatics*, 2:1–78, 2009.

Robert Stalnaker. Pragmatics. *Synthese*, 22:272–289, 1970.

Robert Stalnaker. Presuppositions. *Journal of Philosophical Logic*, 2:447–457, 1973.

Robert Stalnaker. Pragmatic presuppositions. In M. K. Munitz and D. K. Unger, editors, *Semantics and Philosophy*, pages 197–213. NYU, 1974.

Robert Stalnaker. Assertion. In P. Cole, editor, *Syntax and Semantics*. Academic Press, 1978.

Johan van Benthem. *Essays in Logical Semantics*. Springer, 1986.

Johan van Benthem. Semantic parallels in natural language and computation. In *Logic Colloquium '87*. Elsevier, 1989.

Johan van Benthem. *Exploring Logical Dynamics*. CSLI, 1996.

Jan van Eijck and Albert Visser. Dynamic semantics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2010 edition, 2010.

Frank Veltman. Defaults in update semantics. *Journal of Philosophical Logic*, 25: 221–61, 1996.

Kai von Fintel and Anthony Gillies. An opinionated guide to epistemic modality. In Tamar Gendler and John Hawthorne, editors, *Oxford Studies in Epistemology 2*. Oxford, 2007.

Seth Yalcin. Epistemic modals. *Mind*, 116:983–1026, 2007.

Seth Yalcin. Dynamic semantics. In Delia Graff Fara and Gillian Russell, editors, *The Routledge Companion to the Philosophy of Language*. Routledge, forthcoming.