Presupposition and delayed evaluation[1]

Patrick D. Elliott & Martin Hackl

April 23, 2020

[1] 24.979: Topics in semantics
Getting high: Scope, projection, and evaluation order

Schedule

Homework: Next week, after finishing our discussion of presupposition and scope, I'll be talking about the scope of expressive adjectives such as *damn*. Please read chapter 4 of Daniel Gutzmann's 2019 book *The grammar of expressivity*, and send me at least one question by next Wednesday. You can find a `pdf` on the materials section of the class page.

April 30 Presupposition cont.; expressives and scope-taking.

May 7 Expressives cont.; student presentation.

May 14(?) Student presentations.

We'd like to tentatively schedule the make-up class for Thursday May 14, in the usual time slot. Please let us know if this won't work for you, and we'll do our best to come up with a date that works for everyone.

1 Satisfaction and its discontents

1.1 Background on the satisfaction theory

One of the most successful theories of presupposition projection, is the "satisfaction theory".[2]

[2] Heim 1983, Beaver 2001, a.o.

The satisfaction theory is couched in the Stalnakerian perspective (1976, 2002), in which the Common Ground (CG) is modeled as a set of possible worlds – the set of possible worlds compatible with the shared knowledge of the discourse participants.

Informally, the CG is the grand conjunction of all propositions mutually believed to be true by the discourse participants.

Likewise, sentences are taken to denote *propositions* – which can be characterized as a set of possible worlds.

(1) $[\![\text{Ecco swims}]\!] = \{\, w \mid \text{Ecco swims in } w \,\}$

On this theory, when a sentence such as (1) is asserted in a CG c (and the assertion is accepted by the discourse participants), we can easily model the effect of the assertion on the CG via simple set intersection.

In the parlance of the satisfaction theory, we say that the sentence "updates" the CG.

(2) $c \cap \{\, w \mid \text{Ecco swims in } w \,\}$

One of the primary innovations of the satisfaction theory, and dynamic semantics more generally,[3], is to treat the *semantic* contribution of a sentence as an instruction to update a context set:

[3] See especially Heim 1982.

(3) $[\![\text{Ecco swims}]\!] = \lambda c \,.\, c \cap \{\, w \mid \text{Ecco swims in } w \,\}$

In this theory, semantic presupposition receives a natural treatment – a sentence p_π (where π is its presupposition) – requires that π be true throughout the context set c, in order for an update of c can be defined.

We can write this is directly as a definedness condition:

(4) $[\![\text{the dolphin swims}]\!] = \lambda c : c \subseteq \{\, w' \mid \text{there is a dolphin in } w' \,\}$
$\quad . \; c \cap \{\, w \mid \text{a dolphin swims in } w \,\}$

The update in (4) will simply be undefined if it doesn't follow from the mutual beliefs of the discourse participants that *there is a dolphin*.

This captures our qualitative intuitions about the conditions under which a sentence with a presupposition is felicitous.

One of the most beautiful results of the satisfaction theory is that it provides us with the results to capture the Heim-Karttunen projection patterns.

(5) [A dolphin swam] and [the dolphin was fast]. *presuppositionless*

(6) [the dolphin was fast] and [a dolphin swam].
presupposes: there is a dolphin

(7) **Projection in conjunctive sentences**
If A_π, and B_ρ, then a sentence of the form "A and B" presupposes π, and unless A entails ρ, also presupposes ρ

We find an identical pattern in conditional sentences:

(8) If [a dolphin swims], then [the dolphin is fast]. *presuppositionless*

(9) If [the dolphin is fast], then [a dolphin swims].
presupposes: there is a dolphin

(10) **Projection in conditional sentences**
If A_π, and B_ρ, then a sentence of the form "if A then B" presupposes π, and unless A entails ρ, also presupposes ρ

Famously, the way that these patterns are accounted for is to treat natural language *and*, and *if..then...* etc. as operations that manipulate *updates*.

Natural language *and*, for example, can be defined as a connective that takes two updates ϕ and ψ threads them together via function composition – first, the first conjunct updates the context set, then, the second conjunct updates the result.

(11) Dynamic conjunction (def.)
$\phi\ ;\ \psi \coloneqq \psi \circ \phi$

Let's see, informally, how this captures the Heim-Karttunen projection facts.

(12) $[\![\text{a dolphin swam}]\!]\ ;[\![\text{the dolphin was fast}]\!]$
$= \lambda c\ .\ [\lambda c'\ :\ c' \subseteq \{\, w' \mid \text{there's a dolphin in } w' \,\}\ .\ c' \cap \{\, w' \mid \text{a dolphin was fast in } w' \,\}]$
$(c \cap \{\, w \mid \text{a dolphin swam in } w \,\})$

Since first updating *c* with *a dolphin swam*, a subsequent update of the second conjunct is always guaranteed to be defined, just in case the first update is successful. This accurately captured the observed projection pattern.

Since the definition of dynamic conjunction is *asymmetric*, if the first conjunct has a definedness condition, this definedness condition will be inherited by the conjunctive update.

The same kind of story holds for the conditional, although stating the meaning of the dynamic counterpart of material implication is somewhat more complicated.

if p then q is defined essentially via first order equivalence as *it's not the case that p and not q*.

(13) Dynamic conditional (def.)

$$\text{if } \phi\ \psi := \lambda c\ .\ c - (\phi\ c - (\phi\ ;\ \psi)\ c)$$

First, we update c with ϕ, and then subtract the result of doing the sequential update $\phi\ ;\ \psi$ on c. Then we subtract that final result from c.

The component to pay attention to is the fact that the meaning of the operator is stated in terms of discourse sequencing.

This captures the basic Heim-Karttunen projection facts – since the consequent is interpreted in the context of first updating with the antecedent, it is guaranteed to be always defined:

(14) if ⟦a dolphin swam⟧ ⟦the dolphin was fast⟧

$$= \lambda c\ .\ c - \begin{pmatrix} (c \cap \{\, w \mid \text{a dolphin swam in } w \,\}) \\ - ((c \cap \{\, w \mid \text{a dolphin swam in } w \,\}) \cap \{\, w' \mid \text{a dolphin was fast in } w' \,\}) \end{pmatrix}$$

1.2 The proviso problem

The satisfaction theory, while providing an extremely neat explanation for the Heim-Karttunen projection facts, suffers from a well-known deficiency known as the *proviso problem*.[4]

[4] The proviso problem was first brought to light by Geurts 1996.

Consider what we predict as the presupposition of the following sentence:

(15) If a shark swims, then the dolphin was fast.

(16) if ⟦a shark swims⟧ ⟦the dolphin was fast⟧

$$= \lambda c\ .\ c - \begin{pmatrix} (c \cap \{\, w \mid \text{a shark swam in } w \,\}) \\ - \begin{pmatrix} (\lambda c'\ :\ c' \subseteq \{\, w' \mid \text{there's a dolphin in } w' \,\}\ .\ c' \cap \{\, w' \mid \text{a dolphin was fast in } w' \,\}) \\ (c \cap \{\, w \mid \text{a shark swam in } w \,\}) \end{pmatrix} \end{pmatrix}$$

This update will be defined for a context c, just in case c entails that *no shark swam or there is a dolphin*.

That's because, if the global context c has this property, updating it with *a shark swam* is enough to guarantee that the second disjunct must be true – namely, that there is a dolphin, in which case the presupposition of the consequent is guaranteed to be satisfied.

Note that *no shark swam or there is a dolphin* is equivalent to *if a shark swam, then there is a dolphin*, and indeed, the usual way that the proviso problem is stated is as follows: the satisfaction theory predicts weak, *conditional* presuppositions in certain cases.

Here's a classic example:

(17) If Theo hates sonnets, then so does his wife. (Geurts 1996)

In an out-of-the-blue context, we would tend to accommodate (18), not the weaker (19) predicted by, e.g., the satisfaction theory:

(18) *Theo has a wife* attested presupposition

(19) *If Theory hates sonnets then Theo has a wife* predicated presupposition

The implicit assumption here is that, if the presupposition of a sentence p_π isn't entailed by a given context c, we first update c with π.

In other words, we first winnow out worlds from the context where Theo hates sonnets but doesn't have a wife.[5]

The satisfaction theory predicts that if the weaker, conditional statement is part of the common ground, then accommodation will be unnecessary. This seems correct.

(20) We've figured out, that if the butler called in sick on Monday, then someone killed Smith. Furthermore, if the butler called in sick on Monday, it was the butler who killed Smith!
✓We haven't yet figured out whether or not Smith is still alive.

It's only when we have to accommodate that the proviso problem becomes apparent.

(21) We've figured out, that if the butler called in sick on Monday, then it was the butler who killed Smith.
✗We haven't yet figured out whether or not Smith is still alive.

[5] Although we'll only talk about the satisfaction theory here for the sake of exposition, this isn't the only theory that facts the proviso problem. The multidimensional theory has the same problem (Karttunen & Peters 1979), as does the trivalent approach (B. R. George 2007, 2008, Fox 2013).

So the question is, how do we keep the good predictions of the satisfaction theory, without making bad predictions wrt what is accommodated (examples from Mandelkern 2016).

The proviso problem is a problem for other connectives too:

(22) Either Theo doesn't hate sonnets, or he and his wife both hate sonnets.
attested presupposition: *Theo has a wife*
predicted presupposition: *if Theo hates sonnets then Theo has a wife*

A possibly related problem is that the dynamic theory predicts weak projection for triggers embedded under attitude verbs, i.e., (23) is predicted to presuppose that *Alex believes that Robyn used to smoke*[6]

[6] We won't discuss this data today, although Grove does extend his theory to account for these cases.

(23) Alex believes that Robyn stopped smoking.

This is motivated by local satisfaction, since the following sentence is presuppositionless:

(24) If Alex believes that Robyn used to smoke, then he believes that she stopped.

Nevertheless, what is accommodated when (23) is uttered in an out of the blue context is plausibly *that Robyn used to smoke*

(25) Alex believes that Robyn stopped smoking, # but I have no idea if she used to smoke.

1.3 Dismissing a pragmatic response

A disparity between prediction presuppositions and what is accommodated is not necessarily an *insurmountable* problem for the satisfaction theory. Here is the basic idea behind a pragmatic explanation:

(26) *Strengthening*:
For pragmatic reasons, we sometimes accommodate strictly more than is presupposed.

Here is one way of spelling this out (from Mandelkern 2017):

(27) *Plausibility*:

a. When S asserts if p then q_π, her listener compares the relative plausibility of:
   i. S is presupposing $p \supset \pi$
   ii. S is presupposing π

b. S will conclude in favour of (i) iff she has a pragmatic reason to think (it's common ground that) (i) is more plausible than (ii).

This seems to make straightforwardly bad predictions. The following example is from Mandelkern (2016):

(28) ?? John was limping earlier; I don't know why. Maybe he has a stress fracture. I don't know if he plays any sports, but if he has a stress fracture, then he'll stop running cross-country now.

Given the context – the speaker doesn't know if John plays sports – the conditional presupposition predicted by the satisfaction theory: *if John has a stress fracture, he used to run cross-country*, is much more plausible than the unconditional presupposition.

This example, tellingly, becomes OK if we alter the contextual set-up:

(29) John was limping earlier; I don't know why. Maybe he has a stress fracture. If he has a stress fracture, then he'll stop running cross-country now.

Some more problems for a pragmatic account:

Objection from assertion

When we assert "if p the q", why don't we always strengthen to q if q is more plausible?

We need to say something here, e.g., if you knew q, you should have asserted q (wait for the pragmatics block!).

Whatever our account is, it *shouldn't* apply to presupposed content.

Objection from anaphora

Guerts (1996); attributed to van der Sandt:

(30) a. John has a wife; she is a lawyer.
b. ??John is married; she is a lawyer.

Proviso cases pattern with (30a) not (30b):

(31) If Theo hates sonnets, his wife does too. She definitely likes elegies though.

Objection from factives

(32) Walter knows that if Theo hates sonnets, he has a wife.
presupposes: if Theo hates sonnets, then he has a wife

Since this presupposition is identical to that of "If Theo hates sonnets, then his wife does too", why is the latter strengthened and this one not?

Objection from cancellation

If strengthening is pragmatic, it should be cancellable.

(33) If the problem was difficult, then it wasn't Morton who solved it. But as a matter of fact the problem wasn't solved at all.

(34) *We don't know whether Jimbo was murdered or has run away from home. We need to examine his room.*
a. If there are bloodstains in the room, then Jimbo was murdered, and Jimbo's murderer did a sloppy job
b. # If there are bloodstains in the room, then Susie's murderer's did a sloppy job.

1.4 Towards a scopal theory

Grove's strategy in this paper is as follows:

- Start out with a compositional fragment with the resources for dealing with intensionality and alternatives, building on Charlow (2014, 2019).
- Extend this grammar with the resources to deal with presupposition, and sequential update. Show how the proviso problem arises.
- Extend the fragment with mechanisms that allow the evaluation of a presupposition to be *delayed* (scope-taking).

- The proviso problem is resolved by allowing a presupposition trigger to scope out of an environment which would otherwise lead to filtration.

2 *A fragment with alternatives*

In formal semantics, the standard Stalnakerian assumption is that sentences denote sets of possible worlds.

To illustrate, a sentence such as "a dolphin swam" would be assigned the following denotation:

(35) $\{ w \mid \exists x[\mathsf{dolphin}_w\ x \wedge \mathsf{swam}_w\ x] \}$

Throughout the paper, Grove frequently takes advantage of the fact that we can think of *characteristic functions*, as representing sets. We can take (35) to be syntactic sugar for the following function, of type $\mathtt{s} \rightarrow \mathtt{t}$.[7]

(36) $\lambda w\ .\ \exists x[\mathsf{dolphin}_w\ x \wedge \mathsf{swam}_w \mathsf{x}]$ $\mathtt{s} \rightarrow \mathtt{t}$

[7] $\mathtt{s}$ is the type of worlds; $\mathtt{t}$ the type of (bivalent) truth values.

We can think of (36) as a function that takes a world w, and:

- returns $\top$ if w is in (35),
- and $\bot$, if w is not in (35).[8]

[8] Following Grove 2019, we'll write the inhabitants of $\mathtt{t}$ (namely **true** and **false**) as $\top$ and $\bot$.

In other words, the conditions on membership in (35) are *bivalent*.

For reasons that will become clear, Grove adopts a theory which introduces a slight twist on the Stalnakerian formula – rather than sets of possible worlds, sentences will be taken to denote sets of *pairs* of worlds and truth values.

(37) $\{ \langle w, (\mathsf{swam}_w\ x) \rangle \mid \mathsf{dolphin}_w\ x \}$

The meaning in (37) will map $\langle w, \top \rangle$ to $\top$ iff a dolphin swam in w, and $\langle w, \bot \rangle$ to $\top$ iff a dolphin didn't swim in w.

Let's say that we have four worlds: in w_f, flipper but not ecco swam, in f_e, ecco but not flipper swam, in w_{fe} both dolphins swam, and in $w_\varnothing$ no dolphin swam. The extension of (37) will be the following set of pairs:

$$[\![\text{a dolphin swam}]\!] = \left\{\begin{array}{l}\langle w_f, \top\rangle, \langle w_f, \bot\rangle \\ \langle w_e, \top\rangle, \langle w_e, \bot\rangle \\ \langle w_{fe}, \top\rangle, \\ \langle w_\varnothing, \bot\rangle\end{array}\right\}$$

We can think of sentences with indefinites as inducing *indeterminacy* – the sentence "a dolphin" swam has an indeterminate truth value at w, depending on which dolphin in w we have in mind.

We can of course retrieve the classical proposition as follows:

(38) $\{\langle w, (\langle w, \top\rangle \in [\![\text{a dolphin swam}]\!])\rangle\}$

Just as before, we can think of a set of pairs as syntactic sugar for a curried characteristic function, as in (39).

(39) $\lambda wt\,.\,\exists x[\mathsf{dolphin}_w\ x \wedge t = \mathsf{swam}_w\ x]$ $\quad \mathtt{s} \to \mathtt{t} \to \mathtt{t}$

How do we derive these sentential meanings compositionally? Following Charlow, Grove assumes that indefinites introduce alternatives:

(40) $[\![\text{a dolphin}]\!] \coloneqq \{\langle w, x\rangle \mid \mathsf{dolphin}_w\ x\}$ $\quad \mathtt{s} \to \mathtt{e} \to \mathtt{t}$

Taking the four worlds we had before, the extension of (40) would be as follows:

$$\left\{\begin{array}{l}\langle w_f, \mathsf{flipper}\rangle, \langle w_f, \mathsf{ecco}\rangle \\ \langle w_e, \mathsf{flipper}\rangle, \langle w_e, \mathsf{ecco}\rangle \\ \langle w_{fe}, \mathsf{flipper}\rangle, \langle w_{fe}, \mathsf{ecco}\rangle \\ \langle w_\varnothing, \mathsf{flipper}\rangle, \langle w_\varnothing, \mathsf{ecco}\rangle\end{array}\right\}$$

Predicates, on the other hand, are assumed to denote sets of world-predicate pairs. The following entry simply pairs every world w with the predicate that is true of an x is x swam in w.

(41) $[\![\text{swam}]\!] \coloneqq \{\langle w, (\lambda x\,.\,\mathsf{swam}_w\ x)\rangle\}$ $\quad \mathtt{s} \to (\mathtt{e} \to \mathtt{t}) \to \mathtt{t}$

We can compose indefinites and predicates by doing an intensionalized version of Pointwise Function Application (PFA).[9]

[9] In function talk, intensional PFA, which we'll *ap*. This is defined as follows:

(42) $m \overset{\odot}{\mathsf{A}} n$

$$\coloneqq \lambda wp\,.\,\exists x, y \begin{bmatrix} m\ w\ x \\ \wedge\ n\ w\ y \wedge p = x\ \mathsf{A}\ y\end{bmatrix}$$

N.b. that, in defining $\overset{\odot}{\mathsf{A}}$, I depart slightly from Grove who explicitly defines forwards and backwards versions. Under the formulation here, the forwards and backwards variants are implicit in overloaded A.

(43) Ap (def.)

$$m \overset{\odot}{\mathsf{A}}\, n := \{ \langle w, x\ \mathsf{A}\ y \rangle \mid \langle w, x \rangle \in m \wedge \langle w, y \rangle \in n \}$$

$$\mathtt{(s \to (a \to b) \to t) \to (s \to a \to t) \to s \to b \to t}$$

$$\mathtt{(s \to a \to t) \to (s \to (a \to b) \to t) \to s \to b \to t}$$

Now we can compose indefinites and predicates via $\overset{\odot}{\mathsf{A}}$.

$$\{ \langle w, (\mathsf{swam}_w\ x) \rangle \mid \mathsf{dolphin}_w\ x \}$$
$$\overset{\odot}{\mathsf{A}}$$
$$\underset{\text{a dolphin}}{\{ \langle w, x \rangle \mid \mathsf{dolphin}_w\ x \}} \qquad \underset{\text{swam}}{\{ \langle w, (\lambda x\ .\ \mathsf{swam}_w\ x) \rangle \}}$$

Figure 1: Alternative-semantic composition via $\overset{\odot}{\mathsf{A}}$

Not all expressions introduce alternatives – concretely, we need a way of lifting a type $\mathtt{e}$ argument into something that can compose with a predicate via intensional PFA.

We can do this via a polymorphic, intensional variant of Partee's IDENT type shifter, which we'll call *pure*.

(44) Pure (def.)

$$a^{\rho} := \{ \langle w, a \rangle \}$$

Now we can compose a sentence such as *ecco swam*:

$$\{ \langle w, (\mathsf{swam}_w\ \mathsf{ecco}) \rangle \}$$
$$\overset{\odot}{\mathsf{A}}$$
$$\underset{\text{Ecco}^{\rho}}{\{ \langle w, \mathsf{ecco} \rangle \}} \qquad \underset{\text{swam}}{\{ \langle w, (\lambda x\ .\ \mathsf{swam}_w\ x) \rangle \}}$$

Figure 2: Alternative-semantic composition via ap and pure

This meaning pairs each world with either $\top$ or $\bot$ depending on whether Ecco swam in that world.

$$[\![\text{Ecco swam}]\!] = \left\{ \begin{array}{l} \langle w_f, \bot \rangle \\ \langle w_e, \top \rangle \\ \langle w_{fe}, \top \rangle \\ \langle w_{\varnothing}, \bot \rangle \end{array} \right\}$$

Something about the compositional schema we're using here should be familiar from our discussion of continuations. We have, so far, the following ingredients:

- A way of describing meanings that encode both intensionality and indeterminacy – namely, the enriched type-space $\mathtt{s} \to \mathtt{a} \to \mathtt{t}$ (where $\mathtt{a}$ is an ordinary, extensional type).
- A way of doing function application in our enriched type-space – namely, intensional PFA (or *ap*).
- A way of lifting a "normal" meaning into our enriched type-space – namely, *pure*.

This will, hopefully, remind you of how we framed continuation semantics: we had (i) a type constructor $\mathtt{K}$, characterizing the enriched type-space of *scopal* meanings, (ii) SFA for doing function application in the enriched type-space, and (iii) Montague Lift for shifting something normal into a trivially scopal meaning.

This construct is known as an *applicative functor* in the functional programming/category theory literature (Mcbride & Paterson 2008).

Following Grove, we can be more explicit about the applicative functor underlying the fragment we've constructed so far. The enriched type-space we're dealing with is characterized by the type constructor defined in (45).

(45) $\odot\, \mathtt{a} \coloneqq \mathtt{s} \to \mathtt{a} \to \mathtt{t}$

Pure, defined in (46a), is a method for lifting a value into a trivial inhabitant of $\odot$. Ap, defined in (46b), is a method for doing FA in the space characterized by $\odot$. Here, we're giving explicit definitions of these operations, rather than using set talk.

(46) a. $a^{\rho} \coloneqq \lambda w x \,.\, x = a$ $\qquad \mathtt{a} \to \odot\, \mathtt{a}$

b. $m \overset{\odot}{\mathsf{A}} n \coloneqq \lambda w p \,.\, \exists x, y[m\ w\ x\ \wedge n\ w\ y \wedge p = x \mathsf{A}\ y]$

$\odot\,(\mathtt{a} \to \mathtt{b}) \to \odot\, \mathtt{a} \to \odot\, \mathtt{b}$

$\odot\, \mathtt{a} \to \odot\,(\mathtt{a} \to \mathtt{b}) \to \odot\, \mathtt{b}$

3 *Upgrading the fragment to accommodate presupposition*

3.1 *Adding trivalence*

In order to analyze presuppositions, we'll shift to a trivalent setting. Alongside the familiar truth values $\top$ and $\bot$, we'll introduce a new truth value – $\#$.

To model this formally, we'll define a new sum type $\mathtt{t}_{\#}$, the inhabitants of which are the three trivalent truth values. We can think of # as representing a state of uncertainty.

$$\top : \mathtt{t}_{\#}$$
$$\bot : \mathtt{t}_{\#}$$
$$\# : \mathtt{t}_{\#}$$

In order to talk about meaning components which may give rise to undefinedness, Grove makes use of Beaver's δ-operator – this takes an bivalent truth value, and maps $\top$ to itself, and $\bot$ to $\#$.

(47) Beaver's δ-operator (def.)

$$p^{\delta} = \begin{cases} \top & p = \top \\ \# & p = \bot \end{cases} \qquad \delta : \mathtt{t} \to \mathtt{t}_{\#}$$

To briefly illustrate, the following predicate will return # if its argument is a non-dolphin in w:

(48) $\lambda x \,.\, \delta\,(\mathsf{dolphin}_w\; x)$

In set talk, this means that membership conditions on a set can be trivalent – there are three possibilities: (a) x is in X (return $\top$), (b) x is not in X (return $\bot$), or (c) it's undefined whether or not x is in X (return $\#$).

(49) $\{\, x \mid \delta\,(\mathsf{dolphin}_w\; x)\,\}$

In order to simplify the proposal for presupposition projection, Grove assumes a weak Kleene semantics for the metalanguage logical connectives:[10]

Weak Kleene just means that undefinedness *always* projects.

[10] Importantly, Weak Kleene is *not* taken to characterize the meaning of natural language *and*, *if..then..*, etc.

$\wedge$	$\top$	$\bot$	$\#$
$\top$	$\top$	$\bot$	$\#$
$\bot$	$\bot$	$\bot$	$\#$
$\#$	$\#$	$\#$	$\#$

$\rightarrow$	$\top$	$\bot$	$\#$
$\top$	$\top$	$\bot$	$\#$
$\bot$	$\top$	$\top$	$\#$
$\#$	$\#$	$\#$	$\#$

$\rightarrow$	
$\top$	$\bot$
$\bot$	$\top$
$\#$	$\#$

Figure 3: Weak Kleene

Finally, it will be helpful to give a trivalent semantics for the metalanguage existential quantifier. As stated, this semantics gives rise to *existential projection*. In other words, a formula of the form $\exists x[p\ x \wedge \delta(q\ x)]$ is defined iff at least one x is a q.

$\left\{ [\![\phi]\!]^{g'} \,\middle|\, g[x]g' \right\}$	$[\![\ulcorner \exists x\ \phi \urcorner]\!]^{g}$
$\{\top\}$	$\top$
$\{\bot\}$	$\bot$
$\{\#\}$	$\#$
$\{\top, \bot\}$	$\top$
$\{\top, \#\}$	$\top$
$\{\bot, \#\}$	$\bot$
$\{\top, \bot, \#\}$	$\top$

Figure 4: Semantics for existentially quantified formulae

3.2 *Upgrading the applicative functor*

We can now upgrade our old applicative functor $\odot$ into one that can handle not just intensionality and indeterminacy, but also (potential) undefinedness. We'll write this new applicative functor as $\circledast$.

The type constructor is much the same as our old type constructor, only, instead of returning a bivalent truth-value, it returns a trivalent truth-value:

(50) $\circledast\ \mathsf{a} \coloneqq \mathsf{s} \rightarrow \mathsf{a} \rightarrow \mathsf{t}_{\#}$

Here, set talk breaks down somewhat, but we can still talk "as if" (50) characterizes a set of world-value pairs for which membership can be **true**, **false**, or **undefined**. We'll continue to use sets as syntactic sugar for curried functions.

We can now also redefine pure and ap such that they can handle inhabitants of this newly enriched type space:

(51) a. $a^{\rho} \coloneqq \lambda w x\ .\ \delta\ (x = a)$ $\qquad \mathsf{a} \rightarrow \circledast\ \mathsf{a}$

b. $m \overset{\circledast}{\mathsf{A}} n \coloneqq \lambda w p\ .\ \exists x, y[m\ w\ x\ \wedge n\ w\ y \wedge \delta\ (p = x\ \mathsf{A}\ y)]$

$\circledast\ (\mathsf{a} \rightarrow \mathsf{b}) \rightarrow \circledast\ \mathsf{a} \rightarrow \circledast\ \mathsf{b}$

$\circledast\ \mathsf{a} \rightarrow \circledast\ (\mathsf{a} \rightarrow \mathsf{b}) \rightarrow \circledast\ \mathsf{b}$

We now have all of the resources we need to illustrate a simple case of presupposition projection with a definite description.

3.3 *Presupposition projection with definites*

In our current compositional setting, an indefinite such as "a dolphin" takes a world w and an individual x, and:

- returns $\top$ if x is a dolphin in w, and
- $\bot$ if x is not a dolphin in w.

(52) $[\![\text{a dolphin}]\!] \coloneqq \lambda w x \, . \, \mathsf{dolphin}_w \; x$ ⊛ e

In our new, trivalent setting, definites such as "the dolphin" will take a world w, an individual x, and:

- return $\top$ if x is a dolphin in w, and
- return # if x is not a dolphin in w.

(53) $[\![\text{the dolphin}]\!] \coloneqq \lambda w x \, . \, \delta \, (\mathsf{dolphin}_w \; x)$ ⊛ e

We can still use set notation, but as we've mentioned the parallel is obscured somewhat – the result of right-hand side of the set comprehension can be either true, false, or *undefined*:

(54) $[\![\text{the dolphin}]\!] \coloneqq \{ \langle w, x \rangle \mid \delta \, (\mathsf{dolphin}_w \; x) \}$ *set talk*

When we compose the definite description with an ordinary one-place predicate, the result is a function which takes a world w and a (bivalent) truth value t, and returns:

true if there is a dolphin who swims in w, and $t = \top$.

false if there is a dolphin who doesn't swim in w, and $t = \bot$.

undefined if there are no dolphins in w.

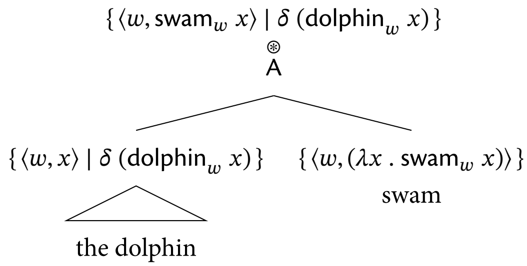

Figure 5: Composition with a definite description

We can still think of the resulting meaning as characterizing a set of world-truth-value pairs, only now, membership in the set may be true, false, or undefined.

We can identify the semantic presupposition of a sentence ϕ as the following set:

(55) The semantic presupposition of ϕ
$\{ w \mid \exists t[(⟦\phi⟧ \ \langle w, t\rangle = \top) \vee (⟦\phi⟧ \ \langle w, t\rangle = \bot)] \}$

This accurately tells us that the semantic presupposition of "the dolphin swam" is the set of worlds in which there is some dolphin – only such worlds paired with a truth value t are mapped to $\top$ or $\bot$.

We've derived the basic presupposition projection properties of definites. The next stage is to develop a theory according to which presuppositions can be *filtered* in certain environments – this will net us the basic results of a satisfaction theory of presupposition.

3.4 Basic presupposition projection

In order to account for the Heim-Karttunen projection pattern, we're going to need a "short-circuited" version of logical conjunction, defined in (6).

&	$\top$	$\bot$	#
$\top$	$\top$	$\bot$	#
$\bot$	$\bot$	$\bot$	$\bot$
#	#	#	#

Figure 6: Short-circuited conjunction

This short-circuited connective is much like ordinary logical conjunction – the difference being that if the first conjunct is **false**, & returns **false**, regardless of the value of the second conjunct.[11]

[11] This is the so-called "middle Kleene" semantics for conjunction. Benjamin R. George (2014) shows that the middle Kleene entries for the truth-functional connectives can be derived from their bivalent entries via a general algorithm; the semantics in (6) need not be stipulated.

If were to imagine that & characterizes the inferences associated with English *and*, this would predict that the following sentence should be judged *false*, rather than undefined (although, to emphasise, we're not taking & not characterize the meaning of *and*).

(56) Trump isn't president and the king of France is bald.

We can now define *discourse sequencing/dynamic conjunction* in terms of &.

(57) Discourse sequencing (def.)
$\phi + \psi \coloneqq \{\langle w, t\rangle \mid \phi\,\langle w, \top\rangle \mathbin{\&} \psi\,\langle w, t\rangle\}$ $\qquad (+) : \circledast\,\mathtt{t} \to \circledast\,\mathtt{t} \to \circledast\,\mathtt{t}$

When we update ϕ with ψ, we take the subset of ψ containing worlds in which ϕ is true.

Let's now illustrate how this emulates the basic predictions of the satisfaction theory of presupposition projection, by taking a concrete example.

(58) A dolphin swam. The dolphin was fast.

We know what each conjunct should denote already:

(59) a. $\{\langle w, \mathsf{swam}_w\ x\rangle \mid \mathsf{dolphin}_w\ x\}$ $\qquad \circledast\,\mathtt{t}$
b. $\{\langle w, \mathsf{fast}_w\ x\rangle \mid \delta\,(\mathsf{dolphin}_w\ x)\}$ $\qquad \circledast\,\mathtt{t}$

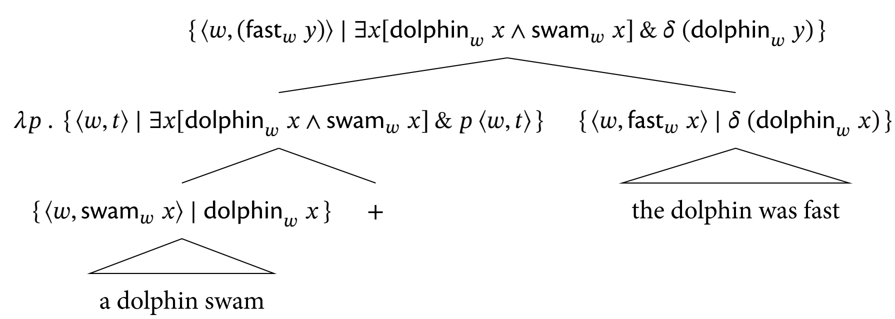

Figure 7: Presupposition filtration in a conjunctive sentence

Remember, we characterize the semantic presupposition of a sentence ϕ as:

$$\{w \mid \exists t[(\llbracket\phi\rrbracket\ \langle w, t\rangle = \top) \vee (\llbracket\phi\rrbracket\ \langle w, t\rangle = \bot)]\}$$

The world truth value pairs which, fed into the conjunctive meaning return either $\top$ or $\bot$, are those worlds in which either (a) there is no dolphin that swam, or (b) there is a dolphin that swam, and is fast.

As noted by Grove – nothing guarantees that, if the conjunctive sentence is true, the dolphin that verifies the first conjunct is the same as the dolphin that verifies the second conjunct.[12]

[12] This is an instantiation of the *binding problem* for presupposition (Karttunen & Peters 1979).

This will be solved in a version of the final analysis enriched with assignments.

3.5 *Encountering the proviso problem*

In order to illustrate the proviso problem, we first need to give a semantics for sentential negation.

(60) Sentential negation (def.)
$\mathsf{not}\ \phi \coloneqq \{\, \langle w, \top \rangle \mid \neg\, (\phi\, \langle w, \top \rangle) \,\}$

Given a proposition with presuppositions ϕ_π, $\mathsf{not}\ \phi$ is a new proposition, such that:

- For any world w, $\langle w, \top \rangle \in \mathsf{not}\ \phi$ just in case $\phi\, \langle w, \top \rangle = \bot$.
- If $\phi\, \langle w, \top \rangle = \#$, then $(\mathsf{not}\ \phi)\, \langle w, \top \rangle = \#$ and $(\mathsf{not}\ \phi)\, \langle w, \bot \rangle = \#$

The consequence is that sentential negation closes off the scope of an indefinite by preventing alternatives from percolating up. To illustrate:

$$\{\, \langle w, \top \rangle \mid \neg\, (\langle w, \top \rangle \in \{\, w, \mathsf{swam}_w\ x \mid \mathsf{dolphin}_w\ x \,\}) \,\}$$

$$\lambda p\ .\ \{\, \langle w, \top \rangle \mid \neg\, (p\, \langle w, \top \rangle) \,\} \qquad \{\, \langle w, \mathsf{swam}_w\ x \rangle \mid \mathsf{dolphin}_w\ x \,\}$$

not

a dolphin swam

Figure 8: Sentential negation closes off indeterminacy

Let's say that we have four worlds: in w_f, flipper but not ecco swam, in f_e, ecco but not flipper swam, in w_{fe} both dolphins swam, and in $w_\varnothing$ no dolphin swam. The extension of "A dolphin swam" will be the following set of pairs:

$$\left\{ \begin{array}{l} \langle w_f, \top \rangle, \langle w_f, \bot \rangle \\ \langle w_e, \top \rangle, \langle w_e, \bot \rangle \\ \langle w_{fe}, \top \rangle, \\ \langle w_\varnothing, \bot \rangle \end{array} \right\}$$

The extension for "A dolphin didn't swim" is the following set of pairs:

$$\{\langle w_{\varnothing}, \top\rangle\}$$

If we have a definite description in the scope of sentential negation, however, the semantic presupposition of the complement is inherited by the negative sentence:

$$\{\langle w, \top\rangle \mid \neg\,(\langle w, \top\rangle \in \{w, \mathsf{swam}_w\ x \mid \delta\,(\mathsf{dolphin}_w\ x)\})\}$$

$$\lambda p\,.\,\{\langle w, \top\rangle \mid \neg\,(p\,\langle w, \top\rangle)\} \qquad \{\langle w, \mathsf{swam}_w\ x\rangle \mid \delta\,(\mathsf{dolphin}_w\ x)\}$$

not

a dolphin swam

Figure 9: Sentential negation allows undefinedness to project

This is because, if there are no dolphins in w, membership of $\langle w, \top\rangle$ in the complement will be undefined, and metalanguage $\neg$ preserves undefinedness (weak Kleene).

We can use this entry for sentential negation to give an entry for the conditional operator. Just as in the classical satisfaction theory, we do so via first order equivalence.

(61) Conditional operator (def.)
$\mathsf{if}\ \phi\ \psi \coloneqq \mathsf{not}\ (\phi + \mathsf{not}\ \psi)$

$\mathsf{if}\ \phi\ \psi$ will turn out true, roughly, if updating ϕ with the negation of ψ turns out false.

Only worlds in which the truth of ϕ guarantees the truth of ψ will remain.

Let's see what this entry for the conditional operator predicts for one of our original sentence used to illustrate the proviso problem:

(62) If Theo has a brother, he'll bring his wetsuit.

The computation of the final meaning is shown in figure (10).

We can more clearly see what the presupposition on the resulting meaning is if we translate the resulting set back into function talk:

(63) $$\lambda wt\,.\,\neg\begin{pmatrix}\mathsf{has\text{-}brother}_w\ \mathsf{Theo} \\ \&\ \neg\,(\exists x[\delta\,(\mathsf{wetsuit}_w\ x) \wedge \mathsf{Theo\ bring}_w\ x]) \wedge t = \top\end{pmatrix}$$

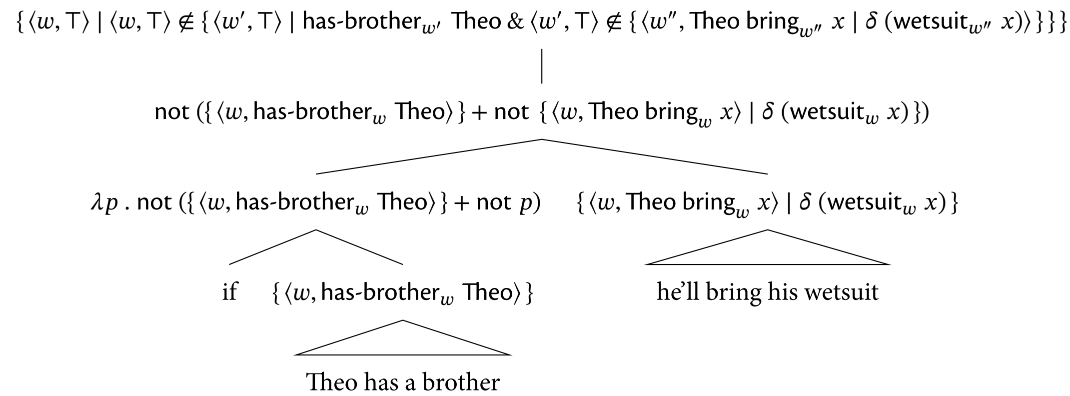

Figure 10: The proviso problem emerges

Since ¬ preserves undefinedness, the presupposition of the second conjunct of & is that Theo has a wetsuit.

The first conjunct asserts that Theo has a brother. By dint on the semantics of &, the presupposition of the second conjunct will only be evaluated in those worlds in which *Theo has a brother* is true.

The definedness condition of the whole sentence is therefore: *Theo has a wetsuit if he has a brother*.

Zooming out, what properties of this fragment are such that the proviso problem arises, and what might we want to tweak in order to avoid it?

In general, the reasons are the following:

- The meaning of the conditional operator is stated in terms of discourse sequencing, the definition of which is motivated by the filtering we observed in conjunctive sentences.

- The presupposition of *his wetsuit* is evaluated within the context of the consequent of the conditional.

As we'll see Grove will seek a way out of this bind by tinkering with the second property of the system – he'll argue that the evaluation of a presupposition can be delayed, via the same mechanisms responsible for delayed evaluation in a more familiar domain – namely, scope.

4 *Shifting perspective: a grammar with scope-taking*

In order make sense of the idea of presuppositional scope, we need to extend our fragment with a new operation: *join*:

(64) Join (def.)
$\mu\ m \coloneqq \{\, \langle w, x\rangle \mid \exists n[\langle w, n\rangle \in m \wedge \langle w, x\rangle \in n]\,\}$ $\qquad \mu : \circledast\,(\circledast\ \mathsf{a}) \rightarrow \circledast\ \mathsf{a}$

Here, m is a set of world-set pairs – join tells us how to take a set of world-set pairs, and "flatten it" into a set of world-value pairs.

Both the main set and the paired sets may, in principle, have definedness conditions on membership.

μ takes m, and gives back a set containing all members of the paired sets in m which preserve the world with which they are paired.

Now, let's see how we convert a definite description into a scope taker.

$[\![\text{the dolphin}]\!] \coloneqq \{\, \langle w, x\rangle \mid \delta\,(\mathsf{dolphin}_w\ x)\,\}$ $\qquad \circledast\ \mathsf{e}$

In order to lift this into a scope-taker, we apply ρ to the contained individual value. We can define an operation, which we'll call *internal lift* which does just this.

(65) Internal lift (def.)
$m^{\upuparrows_{\circledast}} \coloneqq \{\, \langle w, x^{\rho}\rangle \mid \langle w, x\rangle \in m\,\}$ $\qquad \upuparrows_{\circledast} : \circledast\ \mathsf{a} \rightarrow \circledast\,(\circledast\ \mathsf{a})$

Applying internal lift to *the dolphin* gives back a higher-order member of the enriched type-space, where the definedness condition on membership is on the outer layer of the set:

(66) $[\![\text{the dolphin}]\!]^{\upuparrows_{\circledast}} = \{\, \langle w, \{\, \langle w', x\rangle\,\}\rangle \mid \delta\,(\mathsf{dolphin}_w\ x)\,\}$ $\qquad \circledast\,(\circledast\ \mathsf{a})$

In order to compose this with a predicate, the predicate must be lifted via ρ.

We also need a way of doing function application in a *higher-order* enriched type-space. This is defined in the obvious way below:

(67) $m\ \overset{\circledast}{\mathsf{A}}_2\ n \coloneqq \lambda w p\ .\ \exists x, y[m\ w\ x\ \wedge n\ w\ y \wedge \delta\,(p = x\ \overset{\circledast}{\mathsf{A}}\ y)]$
$\circledast\,(\circledast\,(\mathsf{a} \rightarrow \mathsf{b})) \rightarrow \circledast\,(\circledast\ \mathsf{a}) \rightarrow \circledast\,(\circledast\ \mathsf{b})$

$$\circledast\,(\circledast\ \mathsf{a}) \rightarrow \circledast\,(\circledast\,(\mathsf{a} \rightarrow \mathsf{b})) \rightarrow \circledast\,(\circledast\ \mathsf{b})$$

The role of join will be to evaluate the scope of the presupposition trigger. This is illustrated for a trivial example below, in which the presupposition associated with *the dolphin* vacuously takes scope, and is evaluated at the root level.

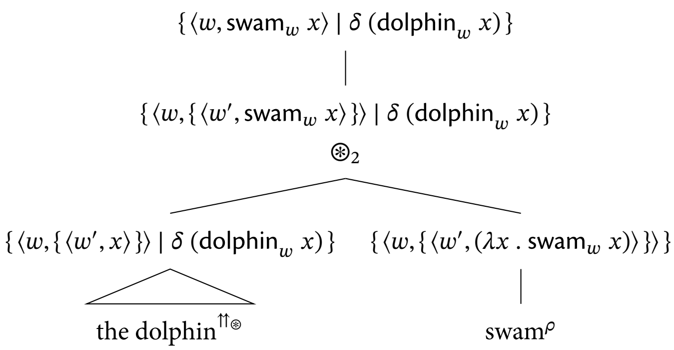

Figure 11: Vacuously scoping a uniqueness presupposition

With this mechanism in hand, however, a presupposition can scope out of an environment in which it would otherwise be filtered.

Now, back to our proviso problem case. We can generate the unconditional presupposition just by applying internal lift to *his wetsuit*, and evaluating via *join* at the root node.

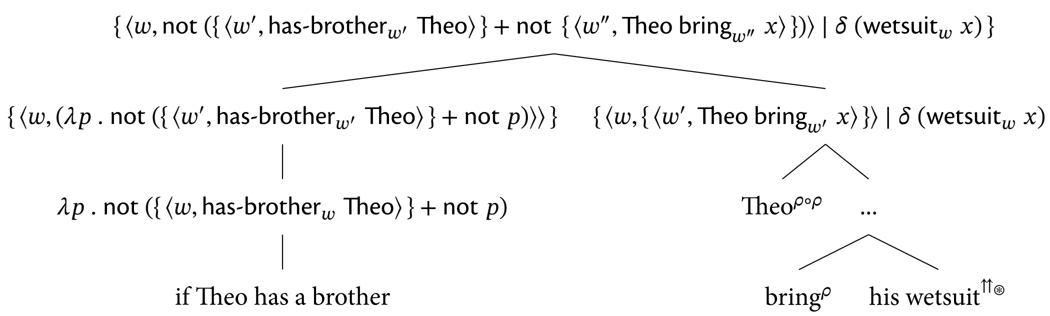

Figure 12: Resolving the proviso problem via scoping out

Applying join to the resulting meaning will have the effect that the presupposition of the outer set takes precedent over either any at-issue content or presuppositions contributed by any inner sets.

Many questions remain:

- Mandelkern's data suggests that, if the presupposition of the consequent isn't entailed in its local context, scoping out is *obligatory*. Why should this be?
- In general, wide-scope seems to be the "default", but as we've discussed in class, scope-shifting operations are often marked in the domain of quantificational scope.

- It can't be quite as simple as that however, since if the presupposition of the consequent *is* entailed in its local context, narrow scope is the default.

Does the following sentence even *have* a reading that presupposes that Theo has a wetsuit? Given hearer charitability, how do we tell?

(68) If Theo is a scuba diver, then he'll bring his wetsuit.

References

Beaver, David I. 2001. *Presupposition and assertion in dynamic semantics* (Studies in logic, language, and information). Stanford, California: CSLI. 314 pp.

Charlow, Simon. 2014. *On the semantics of exceptional scope*. Dissertation.

Charlow, Simon. 2019. The scope of alternatives: indefiniteness and islands. *Linguistics and Philosophy*.

Fox, Danny. 2013. Presupposition projection from quantificational sentences: Trivalence, local accommodation, and presupposition strengthening. In Ivano Caponigro & Carlo Cecchetto (eds.), *From grammar to meaning*. Cambridge: Cambridge University Press.

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*. Ithaca, NY: Cornell University: Linguistic Society of America.

George, Benjamin R. 2014. Some remarks on certain trivalent accounts of presupposition projection. *Journal of Applied Non-Classical Logics* 24(1-2). _eprint: https://doi.org/10.1080/11663081.2014.911521.

Geurts, Bart. 1996. Local satisfaction guaranteed: A presupposition theory and its problems. *Linguistics and Philosophy* 19(3). 259–294.

Grove, Julian. 2019. Satisfaction without provisos. lingbuzz/004914.

Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*. 2011 edition - typesetting by Anders J. Schoubye and Ephraim Glick. University of Massachusetts - Amherst dissertation.

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

Karttunen, Lauri & Stanley Peters. 1979. Conventional implicature. In *Syntax and semantics*, vol. 2.

Mandelkern, Matthew. 2016. Dissatisfaction Theory. *Semantics and Linguistic Theory* 26(0). 391–416.

Mcbride, Conor & Ross Paterson. 2008. Applicative programming with effects. *Journal of Functional Programming* 18(1).

Partee, Barbara. 1986. Noun-phrase interpretation and type-shifting principles. In J. Groenendijk, D. de Jongh & M. Stokhof (eds.), *Studies in discourse representation theory and the theory of generalized quantifiers*, 115–143. Dordrecht: Foris.

Stalnaker, Robert. 1976. Propositions. In A. F. MacKay & D. D. Merrill (eds.), 79–91. New Haven: Yale University Press.

Stalnaker, Robert. 2002. Common Ground. *Linguistics & Philosophy* 25(5/6). 701–721.