

Resolving the proviso problem via scope¹

Patrick D. Elliott & Martin Hackl

April 23, 2020

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

Schedule

Homework: read chapter 4 of Gutzmann's 2019 book *The grammar of expressivity*. 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.

Rescheduled class Student presentations.

1 Satisfaction and its discontents

One of the most successful theories of presupposition projection, is the “satisfaction theory”.²

² Heim 1983

The satisfaction theory suffers from a well-known deficiency known as the *proviso problem*.³

³ The proviso problem was first brought to light by Geurts 1996.

- | | |
|-------------------------|--------------------------------------|
| (1) If p then q_π | \rightsquigarrow if p then π |
| (2) p and q_π | \rightsquigarrow if p then π |

There's doesn't match up with what we tend to accommodate. Imagine an utterance of the following:

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

In an out-of-the-blue context, we would tend to accommodate (4), not the weaker (5) predicted by, e.g., the satisfaction theory:

- (4) Theo has a wife

- (5) If Theory hates sonnets then Theo has a wife

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.

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 multi-dimensional theory has the same problem (Karttunen & Peters 1979), as does the trivalent approach (B. R. George 2007, 2008, Fox 2013).

The *proviso problem* is a problem about *accommodation*

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

- (6) 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.

- (7) We've figured out, that is 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.

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:

- (8) If [Theo hates sonnets and his wife hates sonnets], we shouldn't get Theo a book of sonnets.

↪ *Theo has a wife*

- (9) Either Theo doesn't hate sonnets, or he and his wife both hate sonnets.

↪ *Theo has a wife*

To complicate matters further, sometimes the conditional presupposition seems to make *good* predictions for accommodation:

- (10) If Theo is a scuba-diver, he'll bring his wetsuit on vacation.
- (11) If France is a monarchy, then the king of France is in hiding.

A possibly related problem is that the dynamic theory predicts weak projection for triggers embedded under attitude verbs, i.e., (12) is predicted to presuppose that *Alex believes that Robyn used to smoke*

- (12) Alex believes that Robyn stopped smoking.

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

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

Nevertheless, what is accommodated when (12) is uttered in an out of the blue context is plausibly *that Robyn used to smoke*

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

1.1 A pragmatic response to the proviso problem

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

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

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

- (16) *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 \(2016\)](#):

- (17) ?? 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:

- (18) 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:

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

Proviso cases pattern with (19a) not (19b):

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

Objection from factives

- (21) 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.

- (22) 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.
- (23) *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.

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, using trivalence.

Complete this list

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:

$$(24) \quad \{w \mid \exists x[\text{dolphin}_w x \wedge \text{swam}_w x]\}$$

Throughout the paper, [Grove](#) frequently takes advantage of the fact that we can think of *characteristic functions*, as representing sets. We can take (24) to be syntactic sugar for the following function, of type $s \rightarrow t$.⁴

$$(25) \quad \lambda w . \exists x[\text{dolphin}_w x \wedge \text{swam}_w x] \quad s \rightarrow t$$

⁴ s is the type of worlds; t the type of (bivalent) truth values.

We can think of (25) as a function that takes a world w , and:

- returns \top if w is in (24),
- and \perp , if w is not in (24).⁵

⁵ Following [Grove 2019](#), we'll write the inhabitants of t (namely **true** and **false**) as \top and \perp .

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.

$$(26) \quad \{\langle w, (\text{swam}_w x) \rangle \mid \text{dolphin}_w x\}$$

The meaning in (26) will map $\langle w, \top \rangle$ to \top iff a dolphin swam in w , and $\langle w, \perp \rangle$ 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 w_e , ecco but not flipper swam, in w_{fe} both dolphins swam, and in w_\emptyset no dolphin swam. The extension of (26) will be the following set of pairs:

$$\llbracket \text{a dolphin swam} \rrbracket = \left\{ \begin{array}{l} \langle w_f, \top \rangle, \langle w_f, \perp \rangle \\ \langle w_e, \top \rangle, \langle w_e, \perp \rangle \\ \langle w_{fe}, \top \rangle, \\ \langle w_\emptyset, \perp \rangle \end{array} \right\}$$

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.

Just as before, we can think of a set of pairs as syntactic sugar for a curried characteristic function, as in (27).

$$(27) \quad \lambda w t . \exists x [\text{dolphin}_w x \wedge t = \text{swam}_w x] \quad s \rightarrow t \rightarrow t$$

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

$$(28) \quad \llbracket \text{a dolphin} \rrbracket := \{ \langle w, x \rangle \mid \text{dolphin}_w x \} \quad s \rightarrow e \rightarrow t$$

Taking the four worlds we had before, the extension of (28) would be as follows:

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

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 .

$$(29) \quad \llbracket \text{swam} \rrbracket := \{ \langle w, \langle \lambda x . \text{swam}_w x \rangle \rangle \} \quad s \rightarrow (e \rightarrow t) \rightarrow t$$

We can compose indefinites and predicates by doing an intensionalized version of Pointwise Function Application (PFA).⁶

$$(31) \quad \text{Ap (def.)} \\ m \overset{\odot}{A} n := \{ \langle w, x \overset{\odot}{A} y \rangle \mid \langle w, x \rangle \in m \wedge \langle w, y \rangle \in n \} \\ (s \rightarrow (a \rightarrow b) \rightarrow t) \rightarrow (s \rightarrow a \rightarrow t) \rightarrow s \rightarrow b \rightarrow t \\ (s \rightarrow a \rightarrow t) \rightarrow (s \rightarrow (a \rightarrow b) \rightarrow t) \rightarrow s \rightarrow b \rightarrow t$$

⁶ In function talk, intensional PFA, which we'll *ap*. This is defined as follows:

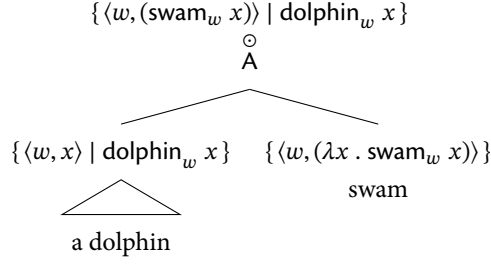
$$(30) \quad m \overset{\odot}{A} n \\ := \lambda w p . \exists x, y [m w x \wedge n w y \wedge p = x \overset{\odot}{A} y]$$

N.b. that, in defining $\overset{\odot}{A}$, 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 $\overset{\odot}{A}$.

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

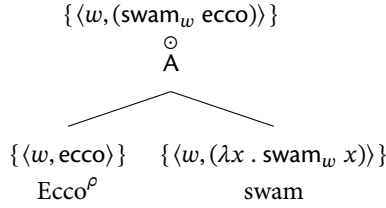
Not all expressions introduce alternatives – concretely, we need a way of lifting a type e argument into something that can compose with a predicate via intensional PFA.

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

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

(32) Pure (def.)
 $a^p := \{\langle w, a \rangle\}$

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

Figure 2: Alternative-semantic composition
via ap and $pure$

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

$$\llbracket \text{Ecco swam} \rrbracket = \left\{ \begin{array}{l} \langle w_f, \perp \rangle \\ \langle w_e, \top \rangle \\ \langle w_{fe}, \top \rangle \\ \langle w_\emptyset, \perp \rangle \end{array} \right\}$$

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 $s \rightarrow a \rightarrow t$ (where 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 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 (33).

$$(33) \quad \odot a := s \rightarrow a \rightarrow t$$

Pure, defined in (34a), is a method for lifting a value into a trivial inhabitant of \odot . Ap, defined in (34b), 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.

$$(34) \quad \begin{array}{ll} \text{a.} & a^P := \lambda wx . x = a & a \rightarrow \odot a \\ & \odot \\ \text{b.} & m \mathbf{A} n := \lambda wp . \exists x, y [m \ w \ x \wedge n \ w \ y \wedge p = x \mathbf{A} y] & \odot (a \rightarrow b) \rightarrow \odot a \rightarrow \odot b \\ & & \odot a \rightarrow \odot (a \rightarrow b) \rightarrow \odot b \end{array}$$

Something about how Julian assumes weak Kleene

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 \perp , we'll introduce a new truth value – $\#$.

To model this formally, we'll define a new sum type $t_{\#}$, the inhabitants of which are the three trivalent truth values. We can think of $\#$ as representing a state of uncertainty regarding the truth of a sentence.

$$\begin{aligned}\top &: \mathbf{t}_{\#} \\ \perp &: \mathbf{t}_{\#} \\ \# &: \mathbf{t}_{\#}\end{aligned}$$

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 \perp to $\#$.

(35) Beaver's δ -operator (def.)

$$p^{\delta} = \begin{cases} \top & p = \top \\ \# & p = \perp \end{cases} \quad \delta : \mathbf{t} \rightarrow \mathbf{t}_{\#}$$

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

(36) $\lambda x . \delta (\text{dolphin}_w x)$

In order to simplify the proposal for presupposition projection, **Grove** assumes a weak Kleene semantics for the metalanguage logical connectives:⁷

\wedge	\top	\perp	$\#$	\rightarrow	\top	\perp	$\#$	\rightarrow	
\top	\top	\perp	$\#$	\top	\top	\perp	$\#$	\top	\perp
\perp	\perp	\perp	$\#$	\perp	\top	\top	$\#$	\perp	\top
$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$

⁷ Importantly, Weak Kleene is *not* taken to characterize the meaning of natural language *and*, *if..then...*, etc.
Figure 3: Weak Kleene

Weak Kleene just means that undefinedness *always* projects.

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 $\exists x[p \wedge \delta(q \ x)]$ is defined iff at least one x is a q .

$\{ \llbracket \phi \rrbracket^{g'} \mid g[x]g' \}$	$\llbracket \exists x \phi \rrbracket^g$
$\{ \top \}$	\top
$\{ \perp \}$	\perp
$\{ \# \}$	$\#$
$\{ \top, \perp \}$	\top
$\{ \top, \# \}$	\top
$\{ \perp, \# \}$	\perp
$\{ \top, \perp, \# \}$	\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 \otimes .

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:

$$(37) \quad \otimes a := s \rightarrow a \rightarrow t_{\#}$$

Here, set talk breaks down, but we can talk as if (37) characterizes a set of world-value pairs for which membership can be **true**, **false**, or **undefined**.

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

$$(38) \quad \begin{array}{ll} \text{a. } a^{\rho} := \lambda w x . \delta (x = a) & a \rightarrow \otimes a \\ \text{b. } m \overset{\otimes}{A} n := \lambda w p . \exists x, y [m w x \wedge n w y \wedge \delta (p = x A y)] & \\ & \otimes (a \rightarrow b) \rightarrow \otimes a \rightarrow \otimes b \\ & \otimes a \rightarrow \otimes (a \rightarrow b) \rightarrow \otimes b \end{array}$$

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
- \perp if x is not a dolphin in w .

$$(39) \quad \llbracket \text{a dolphin} \rrbracket := \lambda w x . \text{dolphin}_w x \quad \otimes e$$

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 .

$$(40) \quad \llbracket \text{the dolphin} \rrbracket := \lambda w x . \delta (\text{dolphin}_w x) \quad \otimes e$$

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

$$(41) \quad \llbracket \text{the dolphin} \rrbracket := \{ \langle w, x \rangle \mid \delta (\text{dolphin}_w x) \} \quad \text{set talk}$$

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 $t = \perp$.

undefined if there are no dolphins in w .

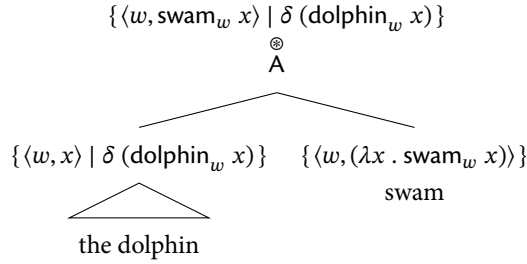


Figure 5: Composition with a definite description

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:

$$(42) \quad \text{The semantic presupposition of } \phi \\ \{ w \mid \exists t [(\llbracket \phi \rrbracket \langle w, t \rangle = \top) \vee (\llbracket \phi \rrbracket \langle w, t \rangle = \perp)] \}$$

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 \perp .

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 presupposition, we need a “short-circuited” version of logical conjunction, defined in (6).

&	T	⊥	#
T	T	⊥	#
⊥	⊥	⊥	⊥
#	#	#	#

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.⁸

If we 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*).

(43) Trump isn’t president and the king of France is bald.

⁸ 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.

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

(44) Discourse sequencing (def.)

$$\phi + \psi := \{ \langle w, t \rangle \mid \phi \langle w, \top \rangle \& \psi \langle w, t \rangle \} \quad (+) : \otimes t \rightarrow \otimes t \rightarrow \otimes t$$

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.

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

We know what each conjunct should denote already:

- (46) a. $\{\langle w, \text{swam}_w x \rangle \mid \text{dolphin}_w x\}$ $\otimes \text{t}$
 b. $\{\langle w, \text{fast}_w x \rangle \mid \delta(\text{dolphin}_w x)\}$ $\otimes \text{t}$

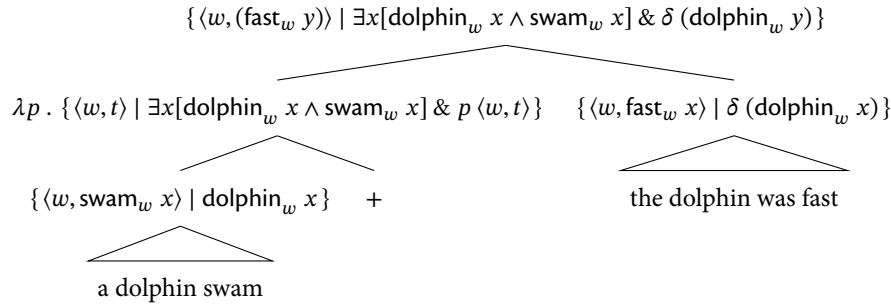


Figure 7: Presupposition filtration in a conjunctive sentence

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

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

The world truth value pairs which, fed into the conjunctive meaning return either \top or \perp , 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.⁹

⁹ 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.

- (47) Sentential negation (def.)
 $\text{not } \phi := \{\langle w, \top \rangle \mid \neg(\phi \langle w, \top \rangle)\}$

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

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

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

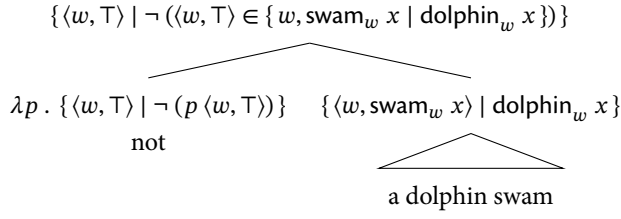


Figure 8: Sentential negation closes off indeterminacy

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

$$\left\{ \begin{array}{l} \langle w_f, T \rangle, \langle w_f, \perp \rangle \\ \langle w_e, T \rangle, \langle w_e, \perp \rangle \\ \langle w_{fe}, T \rangle, \\ \langle w_\emptyset, \perp \rangle \end{array} \right\}$$

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

$$\{ \langle w_\emptyset, T \rangle \}$$

Since there are no presuppositions, the resulting function maps every other world truth-value pair to \perp .

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:

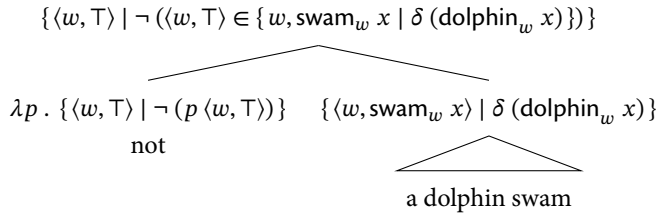


Figure 9: Sentential negation allows undefinedness to project

This is because, if there are no dolphins in w , membership of $\langle w, T \rangle$ 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:

- (48) Conditional operator (def.)
 if $\phi \psi := \text{not} (\phi + \text{not } \psi)$

if $\phi \psi$ will turn out true, roughly, if updating ψ with the negation of ϕ 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 our original sentence used to illustrate the proviso problem:

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

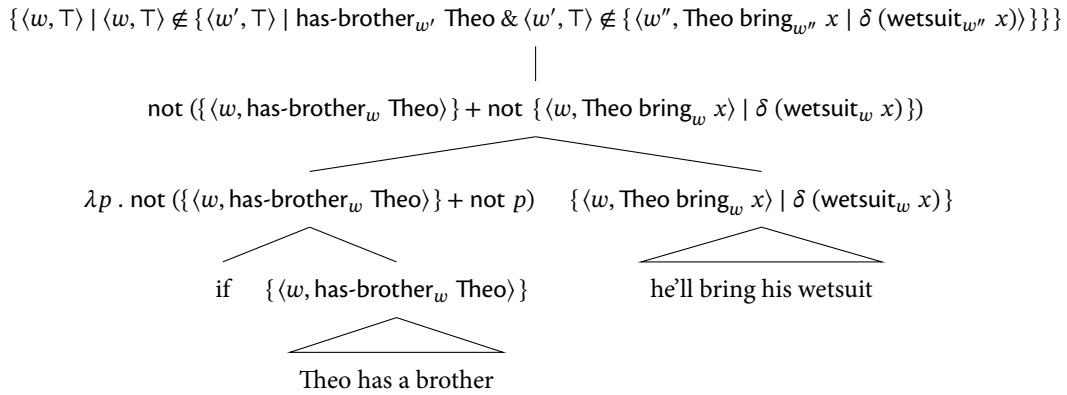


Figure 10: The proviso problem emerges

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

- (50) $\lambda wt . \neg (\text{has-brother}_w \text{ Theo} \& \neg (\exists x [\delta(\text{wetsuit}_w x) \wedge \text{Theo bring}_w x]) \wedge t = T)$

Since \neg 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 of 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.

Semantics for the conditional

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*:

$$(51) \text{ Join (def.)} \quad \mu m := \{ \langle w, x \rangle \mid \exists n [\langle w, n \rangle \in m \wedge \langle w, x \rangle \in n] \} \quad \mu : \otimes (\otimes a) \rightarrow \otimes a$$

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.

$$\llbracket \text{the dolphin} \rrbracket := \{ \langle w, x \rangle \mid \delta (\text{dolphin}_w x) \} \quad \otimes e$$

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.

(52) Internal lift (def.)

$$m^{\uparrow\otimes} := \{ \langle w, x^\rho \rangle \mid \langle w, x \rangle \in m \} \quad \uparrow_{\otimes}: \otimes a \rightarrow \otimes (\otimes a)$$

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:

$$(53) \quad \llbracket \text{the dolphin} \rrbracket^{\uparrow\otimes} = \{ \langle w, \{ \langle w', x \rangle \} \mid \delta(\text{dolphin}_w x) \rangle \} \quad \otimes (\otimes a)$$

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:

$$(54) \quad m \overset{\otimes}{\mathbf{A}}_2 n := \lambda w p . \exists x, y [m w x \wedge n w y \wedge \delta(p = x \overset{\otimes}{\mathbf{A}} y)]$$

$$\otimes (\otimes (a \rightarrow b)) \rightarrow \otimes (\otimes a) \rightarrow \otimes (\otimes b)$$

$$\otimes (\otimes a) \rightarrow \otimes (\otimes (a \rightarrow b)) \rightarrow \otimes (\otimes b)$$

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.

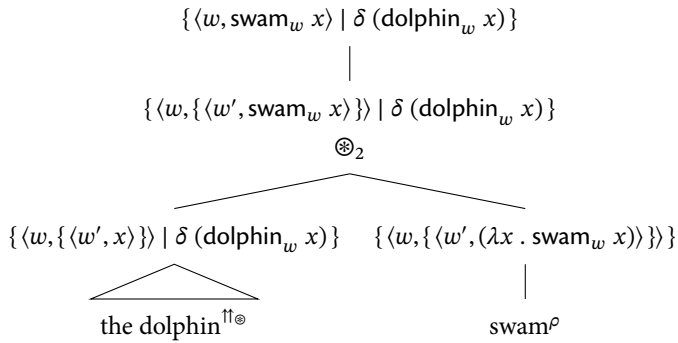


Figure 11: Vacuously scoping a uniqueness presupposition

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.

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.

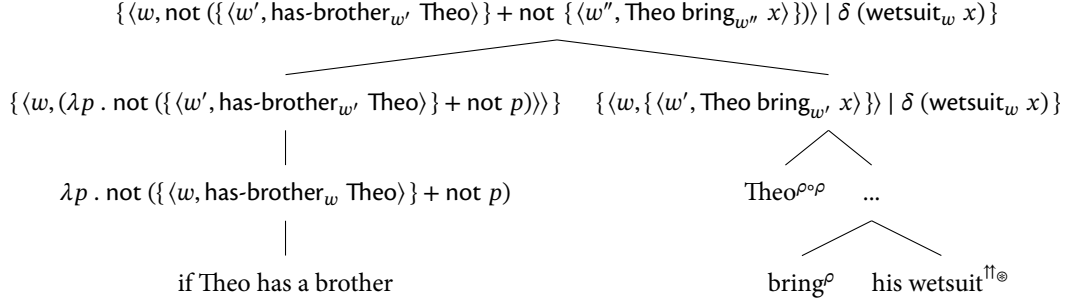


Figure 12: Resolving the proviso problem via scoping out

References

- 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](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. 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.