Satisfaction without provisos

Julian Grove

University of Chicago

SURGE reading group, Rutgers University

December 12, 2019

We have linguistic devices that grammatically encode what we take for granted in making an utterance.

(1) Karlos brought his car.
$\rightsquigarrow$ Karlos has a car. (presupposition)

How do we identify presuppositions?

- Family-of-sentence tests (Chierchia and McConnell-Ginet, 1990)
- “hey, wait a minute!” test (von Fintel, 2004)

Major research question: what grammatical properties of an expression give rise to its presuppositions?

A **compositional** account answers two questions:

- How do we grammatically encode presuppositions in simple expressions (presupposition triggers)?
- How do presuppositions project in complex expressions?
  - The “projection problem” (Langendoen and Savin, 1971)

Today's talk

Outline:

- Investigate an influential compositional framework for studying presupposition projection: "satisfaction theory" (Geurts, 1996).
  - Heim 1983
  - Compositionally derived conditions on dynamic update $\rightsquigarrow$ presuppositions
  - The "proviso problem"
- Build a satisfaction account on top an alternative-semantics for indefinites (Charlow, 2014, 2019a,b)
  - which allows presupposition triggers to *take (exceptional) scope*.
  - Same theory gives rise to exceptional scope for both indefinites and presupposition triggers.
  - Investigate the proviso problem in this setting.
- Suggest how presupposition accommodation might work.

Plan

The satisfaction theory

- Rouch sketch of how it works
- The proviso problem

Alternative semantics and presupposition

- Basic grammar for indefinites
- Adding presupposition
- The proviso problem?

Presupposition accommodation

Plan

- The satisfaction theory
    - Rouch sketch of how it works
    - The proviso problem
- Alternative semantics and presupposition
    - Basic grammar for indefinites
    - Adding presupposition
    - The proviso problem?
- Presupposition accommodation

The satisfaction theory

Rouch sketch of how it works

- Basic ideas come from Heim 1983.
- Sentences denote *context change potentials.*

$$
\begin{aligned}
& [\![\Delta \;\; ; \;\; \textit{it's raining}]\!] \\
= \; & [\![\Delta]\!] + [\![\textit{it's raining}]\!] \\
\text{e.g.,} \quad = \; & \{w \in \mathcal{W} \mid [\![\Delta]\!]^{w} = \top\} \cap \{w \in \mathcal{W} \mid \mathbf{rain}\, w\}
\end{aligned}
$$

- What is $+$? Depends on your (more specific) theory.
    - Might amount to set intersection (for sets of worlds, sets of assignments, etc.).

- What if the sentence updating Δ has a presupposition?

$$[\![\Delta]\!] + [\![\textit{Karlos brought his car}]\!]$$

  - Δ "admits" *Karlos brought his car* only if $[\![\Delta]\!]$ entails that Karlos has a car.
    - "Stalnaker's bridge" (von Fintel, 2008)

- Foregoing assumptions are meant to provide a way of determining what a sentence's presuppositions *are*:
  - S_1 presupposes S_2 iff every context Δ, such that $[\![\Delta]\!] + [\![S_1]\!]$ is successful, entails S_2.
  - *Karlos brought his car* $\rightsquigarrow$ *Karlos has a car*

The satisfaction theory

Rouch sketch of how it works

- Explaining projection behavior: just a matter of using $+$ in the right way.

(2) Karlos has a car, and he brought his car.

- $c + [\![(2)]\!] = (c + [\![\textit{K has a car}]\!]) + [\![\textit{K brought his car}]\!]$
- Update is successful iff each of the individual updates is successful.
- ...iff c entails *If Karlos has a car, then Karlos has a car.* $\rightsquigarrow$ No presuppositions for (2).
- The second sentence's presupposition is *filtered*.

(3) If Karlos has a car, he brought his car.

- $c + [\![(3)]\!]$:
  - $c_1 \coloneqq c + [\![\textit{K has c}]\!]$
  - $c_2 \coloneqq (c + [\![\textit{K has c}]\!]) + [\![\textit{K brought c}]\!]$
  - $= c - (c_1 - c_2)$
- Again, update is successful iff each of the individual updates is. $\rightsquigarrow$ No presuppositions for (3).

Plan

- The satisfaction theory
  - Rouch sketch of how it works
  - The proviso problem
- Alternative semantics and presupposition
  - Basic grammar for indefinites
  - Adding presupposition
  - The proviso problem?
- Presupposition accommodation

- Geurts 1996: there's a big problem!

(4) It's raining, and my car is too far away.

- Update successful iff each of the individual updates is.
  - $c + [\![\textit{it's raining, and my car is too far away}]\!]$
    $= (c + [\![\textit{it's raining}]\!]) + [\![\textit{my car is too far away}]\!]$
- …iff the context entails *if it's raining, I have a car.*
- (4) $\rightsquigarrow$ *if it's raining, I have a car* ☹

(5) If EWR is nearby, I can pick my sister up once she lands.

- Individual updates:
  - $c_1 \coloneqq c + [\![\textit{EWR is nearby}]\!]$
  - $c_2 \coloneqq (c + [\![\textit{EWR is nearby}]\!]) + [\![\textit{pick sister up}]\!]$
  - $= c - (c_1 - c_2)$
- (5) $\rightsquigarrow$ *if EWR is nearby, I have a sister* ☹

- According to the satisfaction theory, filtration is *automatic*.
- But sometimes it shouldn't happen.

Plan

- Let's build our satisfaction account by starting with a grammar in which indefinites systematically give rise to *alternatives*.
- Importing basic ideas from Charlow 2014.
- I'll add in a few extra things though:
  - intensionality (we want to compose contexts)
  - Quantifier Raising & Predicate Abstraction(Heim and Kratzer, 1998)
    - Just for expository purposes (see, e.g., Charlow, 2019a,b).
  - presupposition

- Basic ingredients:
  - $[\![\textit{a dolphin}]\!] = \{\langle w, x\rangle \mid \mathbf{dolphin}\, wx\}$
    - Really, $(\lambda w, x.\mathbf{dolphin}\, wx)$.
    - Type $s \to e \to t$ (shorthand: $\mathbf{S}(e)$).
  - $[\![\textit{swam}]\!] = (\lambda y.\{\langle w, \mathbf{swam}\, wy\rangle \mid w \in \mathcal{W}\})$
    - Really, $(\lambda y, w, t.t = \mathbf{swam}\, wy)$.
    - Type $e \to s \to t \to t$ (shorthand: $e \to \mathbf{S}(t)$).
- What we want:
  $[\![\textit{a dolphin swam}]\!] = \{\langle w, \mathbf{swam}\, wx\rangle \mid \mathbf{dolphin}\, wx\}$
  - Type $\mathbf{S}(t) = s \to t \to t$ (a “proposition”).
  - If we want to know if ϕ true at some world w, we just ask: “is $\langle w, \top\rangle \in \phi$?”

- To get there, we need to somehow compose something of type $\mathbf{S}(e)$ (*a dolphin*) with something of type $e \to \mathbf{S}(t)$ (*swam*).
- It would be nice if we could turn *a dolphin* into something of type $(e \to \mathbf{S}(t)) \to \mathbf{S}(t)$.
  - A *quantifier*: takes scope over functions into propositions, to return a proposition.
- Following Charlow (2019a,b), let's use an operator $(\cdot)^{\overset{\mathbf{s}}{\ggcurly}}$ ('bind') to do the trick.

$$(\cdot)^{\overset{\mathbf{s}}{\ggcurly}} : \mathbf{S}(\alpha) \to (\alpha \to \mathbf{S}(\beta)) \to \mathbf{S}(\beta)$$

$$m^{\overset{\mathbf{s}}{\ggcurly}} k = \{\langle w, y\rangle \mid (\exists x.\langle w, x\rangle \in m \wedge \langle w, y\rangle \in kx)\}$$

$$\begin{aligned} & \{\langle w, x\rangle \mid \mathbf{dolphin}\, wx\}^{\overset{\mathbf{s}}{\ggcurly}} k \\ = \; & \{\langle w, t\rangle \mid (\exists x.\mathbf{dolphin}\, wx \wedge \langle w, t\rangle \in kx)\} \end{aligned}$$

- *A dolphin swam*:

$$\{\langle w, \textbf{swam}\, wx\rangle \mid \textbf{dolphin}\, wx\}$$

$$\mid =$$

$$\{\langle w, t\rangle \mid (\exists x.\textbf{dolphin}\, wx \ \wedge\ \langle w, t\rangle \in \{\langle w, \textbf{swam}\, wx \mid w \in \mathcal{W}\})\}$$

$$\{\langle w, x\rangle \mid \textbf{dolphin}\, wx\}^{\overset{S}{\gg}} \qquad (\lambda y.\{\langle w, \textbf{swam}\, wy\rangle \mid w \in \mathcal{W}\})$$

swam

$$\mid \overset{S}{\gg}$$

$$\{\langle w, x\rangle \mid \textbf{dolphin}\, wx\}$$

a dolphin

Plan

- So far, we've previewed a grammar for composing propositions of type $\mathbf{S}(t) = s \to t \to t$.
- Let's add in presupposition:
  - A new type, $t_{\#}$, inhabitted by three truth values:
    - $\top : t_{\#}$ 'true'
    - $\bot : t_{\#}$ 'false'
    - $\# : t_{\#}$ 'undefined'
  - Keep old type t, too ($\top : t$ and $\bot : t$, but $\# \not: t$).
- Next, some preliminaries: interpreting some useful constants (bear with me).

- We need a new semantics for connectives!
  -

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

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

$\neg$	
$\top$	$\bot$
$\bot$	$\top$
#	#

[1]

  - $(\phi \wedge \#) = (\# \wedge \phi) = \#$.
  - $(\phi \rightarrow \#) = (\# \rightarrow \phi) = \#$.
  - $\neg\# = \#$.
  - Undefinedness automatically projects.

[1] The "internal" logic of Bochvar 1939; a.k.a. "weak Kleene".

- We also need a new semantics for the existential quantifier!
$\{[\![\phi]\!]_{\mathcal{M},g'} \mid g[x]g'\}$	$[\![\ulcorner(\exists x.\phi)\urcorner]\!]_{\mathcal{M},g}$
$\{\top\}$	$\top$
$\{\bot\}$	$\bot$
$\{\#\}$	$\#$
$\{\top,\bot\}$	$\top$
$\{\top,\#\}$	$\top$
$\{\bot,\#\}$	$\bot$
$\{\top,\bot,\#\}$	$\top$
  - Undefined *only when the only potential witnesses lead to undefinedness.*
  - If there's any witness at all, make it true.
  - If there are no witnesses, but some potential ones lead to falsity, make it false.
  - Undefinedness is a last resort.

- Last important piece: the δ-operator (Beaver, 1999, 2001; Beaver and Krahmer, 2001; Coppock and Beaver, 2015, i.a.).
  - How it works:

$$\delta : t \rightarrow t_{\#}$$

$$\delta\top = \top$$

$$\delta\bot = \#$$

  - Say you've got $\ulcorner \phi \wedge \delta\psi \urcorner$:
    - true or false, depending on ϕ (assuming it's defined)
    - defined or undefined, depending on ψ
  - Turns a truth condition into a definedness condition.

- Those are all the basic pieces we need.
- Before, our propositions were of type $\mathbf{S}(t)(= s \to t \to t)$.
- Let's upgrade them to our new setting:

$$\{\langle w, \mathbf{swam}\, wx\rangle \mid \mathbf{dolphin}\, wx\} : \mathbf{P}(t)$$

$$= s \to t \to t_{\#}$$

Really, $(\lambda w, t.\mathbf{dolphin}\, wx \wedge t = \mathbf{swam}\, wx) : \mathbf{P}(t)$

- For any $\phi : \mathbf{P}(t)$,
  - ϕ is *true* at w iff $\langle w, \top\rangle \in \phi$;
  - ϕ is *false* at w iff $\langle w, \top\rangle \notin \phi$;
  - ϕ is *undefined* at w iff it can't be said whether or not $\langle w, \top\rangle \in \phi$ (i.e., if $\phi\, w \top = \#$).
- Presuppositions of ϕ: set of worlds at which ϕ is defined.

- New basic ingredients:
  - $[\![\textit{a dolphin}]\!] = \{\langle w, x\rangle \mid \textbf{dolphin}\, wx\}$
    - Really, $(\lambda w, x.\textbf{dolphin}\, wx) : \mathbf{P}(e)$.
  - $[\![\textit{swam}]\!] = (\lambda y.\{\langle w, \textbf{swam}\, wy\rangle \mid w \in \mathcal{W}\})$
    - Really, $(\lambda y, w, t.t = \textbf{swam}\, wy) : e \to \mathbf{P}(t)$.
  - $[\![\textit{the dolphin}]\!] = \{\langle w, x\rangle \mid \delta(\textbf{dolphin}\, wx)\}$
    - Really, $(\lambda w, x.\delta(\textbf{dolphin}\, wx)) : \mathbf{P}(e)$.
- What we want:

$$\begin{aligned}[\![\textit{the dolphin swam}]\!] &= \{\langle w, \textbf{swam}\, wx\rangle \mid \delta(\textbf{dolphin}\, wx)\} \\ &= (\lambda w, t.(\exists x.\delta(\textbf{dolphin}\, wx) \wedge t = \textbf{swam}\, wx))\end{aligned}$$

  - $\langle w, \top\rangle \in [\![\textit{the dolphin swam}]\!]$ iff a dolphin swam in w.
  - If no dolphin in w, then we can't check this (i.e., we get #).
  - The worlds w in which $\langle w, \top\rangle$ can be checked for membership are all dolphin worlds… *existence presupposition*!

- We'll invoke the type shift $(\cdot)^{\overset{\mathbf{P}}{\ggg}}$ to turn our *presupposition trigger* into a quantifier.

$$(\cdot)^{\overset{\mathbf{P}}{\ggg}} : \mathbf{P}(\alpha) \to (\alpha \to \mathbf{P}(\beta)) \to \mathbf{P}(\beta)$$
$$m \overset{\mathbf{P}}{\ggg} k = \{\langle w, y\rangle \mid (\exists x.\langle w, x\rangle \in m \wedge \langle w, y\rangle \in kx)\}$$

$$\{\langle w, x\rangle \mid \delta(\mathbf{dolphin}\, wx)\} \overset{\mathbf{P}}{\ggg} k$$
$$= \{\langle w, t\rangle \mid (\exists x.\delta(\mathbf{dolphin}\, wx) \wedge \langle w, t\rangle \in kx)\}$$

- Once fed its scope, the presupposition trigger creates a proposition with presuppositions.

Alternative semantics and presupposition

Adding presupposition

- *The dolphin swam*:

$$\{\langle w, \mathbf{swam}\, wx\rangle \mid \delta(\mathbf{dolphin}\, wx)\}$$

$$\Big| =$$

$$\{\langle w, t\rangle \mid (\exists x.\delta(\mathbf{dolphin}\, wx) \ \wedge \ \langle w, t\rangle \in \{\langle w, \mathbf{swam}\, wx \mid w \in \mathcal{W}\})\}$$

$$\{\langle w, x\rangle \mid \delta(\mathbf{dolphin}\, wx)\}^{\overset{S}{\gg=}} \qquad (\lambda y.\{\langle w, \mathbf{swam}\, wy\rangle \mid w \in \mathcal{W}\})$$

swam

$$\Big| \overset{S}{\gg=}$$

$$\{\langle w, x\rangle \mid \delta(\mathbf{dolphin}\, wx)\}$$

the dolphin

Plan

(2) If Karlos has a car, he brought his car .

- Basic pieces:
  - $[\![\textit{Karlos has a car}]\!] = \{\langle w, \mathbf{have}\, wx\rangle \mid \mathbf{car}\, wx\}$
  - $[\![\textit{Karlos brought his car}]\!] =$ $$\{\langle w, \mathit{brought}\, wx\mathbf{k}\rangle \mid \delta(\mathbf{car}\, wx \wedge \mathbf{have}\, wx)\}$$
  - *if*…
    - A new connective, $\ulcorner\Rightarrow\urcorner$:

$\Rightarrow$	$\top$	$\bot$	$\#$
$\top$	$\top$	$\bot$	$\#$
$\bot$	$\top$	$\top$	$\top$
$\#$	$\#$	$\#$	$\#$

    - $(\bot \Rightarrow \#) = \top$
    - Only checks definedness of the consequent if the antecedent is true.
  - $$\mathsf{if} : \mathbf{P}(t) \to \mathbf{P}(t) \to \mathbf{P}(t)$$ $$\mathsf{if}\,\phi\psi \coloneqq \{\langle w, \top\rangle \mid \langle w, \top\rangle \in \phi \Rightarrow \langle w, \top\rangle \in \psi\}$$

- *If Karlos has a car, he brought his car*:

$$\{\langle w, \top\rangle \mid (\exists x.\mathbf{car}\,wx \wedge \mathbf{have}\,wx\mathbf{k}) \Rightarrow (\exists x.\delta(\mathbf{car}\,wx \wedge \mathbf{have}\,wx\mathbf{k}) \wedge \mathbf{brought}\,wx\mathbf{k})\}$$

$$(\lambda\psi.\{\langle w, \top\rangle \mid (\exists x.\mathbf{car}\,wx \wedge \mathbf{have}\,wx\mathbf{k}) \Rightarrow \langle w, \top\rangle \in \psi\})$$

$$\{\langle w, \mathbf{brought}\,wx\mathbf{k}\rangle \mid \delta(\mathbf{car}\,wx \wedge \mathbf{have}\,wx)\}$$

K brought his car

if

if

$$\{\langle w, \mathbf{have}\,wx\rangle \mid \mathbf{car}\,wx\}$$

K has a car

- $\{\langle w, \top\rangle \mid (\exists x.\mathbf{car}\,wx \wedge \mathbf{have}\,wx\mathbf{k}) \Rightarrow (\exists x.\delta(\mathbf{car}\,wx \wedge \mathbf{have}\,wx\mathbf{k}) \wedge \mathbf{brought}\,wx\mathbf{k})\}$
  $= \{\langle w, \top\rangle \mid (\exists x.\mathbf{car}\,wx \wedge \mathbf{have}\,wx\mathbf{k}) \Rightarrow (\exists x.\mathbf{car}\,wx \wedge \mathbf{have}\,wx\mathbf{k} \wedge \mathbf{brought}\,wx\mathbf{k})\}$

- $\langle w, \top\rangle$ is in this set iff Karlos brought a car he has in w, as long as he has a car in w.

- Definedness conditions are trivial: *filtration!*

Alternative semantics and presupposition

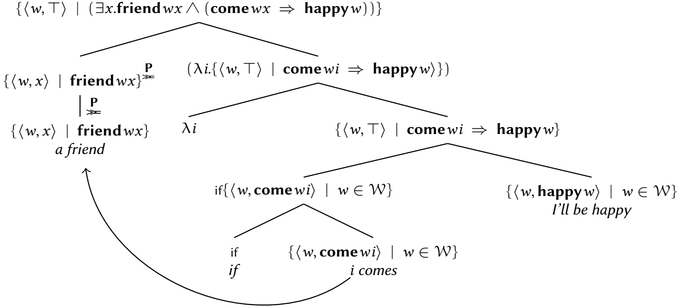
The proviso problem?

- *If EWR is nearby, I'll get my sister*:

  - $\{\langle w, \top\rangle \mid \mathbf{nearby}\, w \Rightarrow (\exists x.\delta(\mathbf{sis}\, w x \mathbf{me}) \wedge \mathbf{get}\, w x \mathbf{me})\}$
    - $(\lambda\psi.\{\langle w, \top\rangle \mid \mathbf{nearby}\, w \Rightarrow \langle w, \top\rangle \in \psi\})$
      - if — *if*
      - $\{\langle w, \mathbf{nearby}\, w\rangle \mid w \in \mathcal{W}\}$ — *ERW is nearby*
    - $\{\langle w, \mathbf{get}\, w x \mathbf{me}\rangle \mid \delta(\mathbf{sis}\, w x \mathbf{me})\}$ — *I'll get my sister*

- $\{\langle w, \top\rangle \mid \mathbf{nearby}\, w \Rightarrow (\exists x.\delta(\mathbf{sis}\, w x \mathbf{me}) \wedge \mathbf{get}\, w x \mathbf{me})\}$
- $\langle w, \top\rangle$ can be checked for membership in this set iff:
  - $\mathbf{nearby}\, w \Rightarrow (\exists x.\mathbf{sis}\, w x \mathbf{m})$
- Should presuppose *If EWR nearby, I have a sister.* ☹

- Our framework's analog of the proviso problem:
    - Filtration is built into the semantics of the conditional (and it should be—look at (2)!).
    - But then the presupposition of (5) is filtered, which we don't want.

- Quick detour: indefinites.
- What do we know about indefinites?
  - They take exceptional scope!
  - *If a friend of mine comes, I'll be happy* $\qquad$ (*a friend* $>$ *if*)
  - 

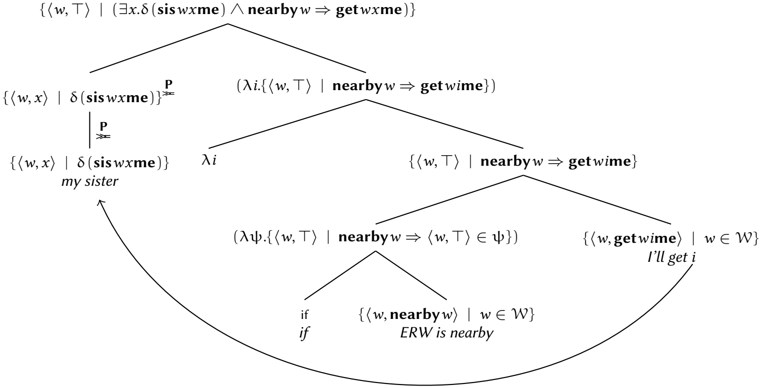
- Somehow, our theory has to allow the indefinite to scope out of the conditional (see Charlow 2019a,b).

- But if the theory allows indefinites to do this…
  then it also allows presupposition triggers to do it!
- End of detour.

Alternative semantics and presupposition

The proviso problem?

- *If EWR is nearby, I'll get my sister* (take 2):



- $\{\langle w, \top\rangle \mid (\exists x.\delta(\textbf{sis}\, w x \textbf{me}) \wedge \textbf{nearby}\, w \Rightarrow \textbf{get}\, w x \textbf{me})\}$
- $\langle w, \top\rangle$ can be checked for membership in this set iff:
  - $(\exists x.\textbf{sis}\, w x \textbf{me})$
- Should presuppose *I have a sister.* ☺

- If we admit exceptional scope for indefinites, then we admit exceptional scope for presupposition triggers automatically.
- But once we do the latter, we may circumvent the proviso problem.

Plan

The satisfaction theory
- Rouch sketch of how it works
- The proviso problem

Alternative semantics and presupposition
- Basic grammar for indefinites
- Adding presupposition
- The proviso problem?

Presupposition accommodation

Presupposition accommodation

- Lewis (1979) observes that presuppositions which are not satisfied by prior discourse are often *accommodated*.

(6) I lost my gloves.

- Heim (1983) further observes that examples in which presuppositions are *cancelled* might receive an analysis in terms of an interpretation strategy she calls “local accommodation”.

(7) The dolphin wasn’t fast, because there is no dolphin!

  - No existence presupposition.
  - Basic idea: the presupposition is promoted to at-issue content, which is then targeted by the negation.

Presupposition accommodation

- To describe sentences like (7), we could introduce an operator, δ^{-1}, which promotes definedness conditions back to truth conditions.
  - $\delta^{-1}\# = \bot$
  - $\delta^{-1}\top = \top$ and $\delta^{-1}\bot = \bot$.
- In terms of it, we may define a new type shift:

$$\text{accom} : \mathbf{P}(\alpha) \to \mathbf{P}(\alpha)$$
$$\text{accom}\, m \coloneqq (\lambda w, x.\delta^{-1}(m w x))$$

$$\begin{aligned} & \text{accom}\{\langle w, x\rangle \mid \delta(\mathbf{dolphin}\, wx)\} \\ = {} & \{\langle w, x\rangle \mid \delta^{-1}(\delta(\mathbf{dolphin}\, wx))\} \\ = {} & \{\langle w, x\rangle \mid \mathbf{dolphin}\, wx\} \end{aligned}$$

  - From the meaning of a definite description, we get the meaning of the corresponding indefinite!

Presupposition accommodation

- If such a type shift could apply in (7), then (7) would be semantically equivalent to (8) (with the indefinite taking narrow scope).

(8) A dolphin wasn't fast, because there is no dolphin!

- If an analysis like this were viable, then it would allow us to view accommodation as simple model-theoretic operation.

Conclusion

- We should really try to hold onto the satisfaction account of presuppositin projection because it allows us to describe presupposition projection using nothing more than the tools semanticists are used to (and need anyway).
  - Truth conditions (sets of worlds)
  - Compositional view of the syntax-semantics interface
- It has been hindered by the proviso problem, but…
  - the problem is overcome if we state the account in terms Charlow's independently motivated theory of indefinites.

References

Beaver, David, and Emiel Krahmer. 2001. A partial account of presupposition projection. *Journal of Logic, Language and Information* 10:147–182. URL https://doi.org/10.1023/A:1008371413822.

Beaver, David I. 1999. Presupposition accomodation: A plea for common sense. In *Logic, language, and computation*, ed. Lawrence S. Moss, Jonathan Ginzburg, and Rijke de Maarten, volume 2, 21–44. Stanford: CSLI Publications.

Beaver, David I. 2001. *Presupposition and assertion in dynamic semantics*. Studies in Logic, Language and Information. Stanford: CSLI Publications. URL https://semanticsarchive.net/Archive/jU1MDVmZ.

Bochvar, Dmitry A. 1939. On a three valued calculus and its application to the analysis of contradictories. *Matematicheskii Sbornik* 4:287–308. URL https://doi.org/10.2307/226908.

Charlow, Simon. 2014. On the semantics of exceptional scope. Doctoral Dissertation, NYU, New York. URL https://semanticsarchive.net/Archive/2JmMWRjY.

References

Charlow, Simon. 2019a. The scope of alternatives: indefinites and islands. *Linguistics and Philosophy* 1–46. URL https://doi.org/10.1007/s10988-019-09278-3.

Charlow, Simon. 2019b. Static and dynamic exceptional scope. URL https://ling.auf.net/lingbuzz/004650, forthcoming in *Journal of Semantics*.

Chierchia, Gennaro, and Sally McConnell-Ginet. 1990. *Meaning and grammar: An introduction to semantics*. Cambridge: MIT Press.

Coppock, Elizabeth, and David Beaver. 2015. Definiteness and determinacy. *Linguistics and Philosophy* 38:377–435. URL https://doi.org/10.1007/s10988-015-9178-8.

von Fintel, Kai. 2004. Would you believe it? the king of France is back! presuppositions and truth-value intuitions. In *Descriptions and beyond*, ed. Marga Reimer and Anne Bezuidenhout, chapter 8, 269–296. Oxford: Oxford University Press.

References

von Fintel, Kai. 2008. What is presupposition accommodation, again? *Philosophical Perspectives* 22:137–170. URL https://doi.org/10.1111/j.1520-8583.2008.00144.x.

Geurts, Bart. 1996. Local satisfaction guaranteed: A presupposition theory and its problems. *Linguistics and Philosophy* 19:259–294. URL https://doi.org/10.1007/BF00628201.

Heim, Irene. 1983. On the projection problem for presuppositions. In *Proceedings of the 2nd West Coast Conference on Formal Linguistics*, ed. Michael D. Barlow, Daniel P. Flickinger, and Nancy Wiegand, 114–125. Stanford: Stanford University Press.

Heim, Irene, and Angelika Kratzer. 1998. *Semantics in generative grammar*. Malden: Blackwell.

Langendoen, Donald Terence, and Harris B. Savin. 1971. The projection problem for presuppositions. In *Studies in linguistic semantics*, ed. Charles J.. Fillmore and Donald Terence Langendoen, 55–60. Holt, Rinehart & Winston.

References

Lewis, David. 1979. Scorekeeping in a language game. In *Semantics from different points of view*, ed. Rainer Bäurele and Arnim von Stechow, volume 6 of *Springer Series in Language and Communication*, 172–187. Berlin: Springer. URL https://doi.org/10.1007/978-3-642-67458-7_12.