The satisfaction theory

24.954: Pragmatics in Linguistic Theory

Patrick D. Elliott

September 2, 2019

1 Introduction

1.1 Readings

- Irene Heim. 1983. On the projection problem for presuppositions. In *Proceedings of WCCFL 2*, 114–125. Stanford University
- Kadmon (1990) *Formal Pragmatics*, chapter 6.
- Rothschild 2011?
- Get them to read some Chierchia

1.2 Notational conventions

- λxy . ... is to be read as λx . λy . ...

Be precise about other things here

2 Dynamic Semantics

2.1 Context Change Potentials

A Context Change Potential (CCP) is a (possibly partial) function over contexts. For example, the CCP associated with the utterance "Paul vapes" is:

(1) $\lambda c \,.\, \lambda w \,.\, c\ w \wedge \mathsf{p\ vapes}_w$

Assuming that sentences denote *propositions*, how do we get from an ordinary (static) sentential meaning, to a CCP? Let's define an operator Assert ($\mathbb{A}$) to accomplish this for us:

(2) Assert (first attempt)
$\mathbb{A}\ p \coloneqq \lambda c \,.\, \lambda w \,.\, c\ w \wedge p\ w$

(3) $\mathbb{A}\ (\llbracket \text{Paul vapes} \rrbracket) = (1)$

Suppose that the context set contains both worlds in which Paul vapes, and worlds in which he doesn't, representing a state of ignorance wrt whether or not Paul vapes, i.e.:

(4) $c_1 = \lambda w \,.\, \mathsf{p\ vapes}_w \vee \neg\, \mathsf{p\ vapes}_w$

We update the context set c_1 with the CCP associated with "Paul vapes" by *applying* the CCP to the context set. The result is an *updated* context set which only includes worlds in which paul vapes.

(5) $[\lambda c \,.\, \lambda w \,.\, c\ w \wedge \mathsf{p\ vapes}_w]\ c_1 = \lambda w \,.\, \mathsf{p\ vapes}_w$

Note that we can always get back from a CCP to a proposition. Let's define an operator $\downarrow$ to do this for us:

Check that this works out

(6) $d^{\downarrow} \coloneqq \lambda w' \,.\, (d\ (\lambda w \,.\, w = w')) = (\lambda w \,.\, w = w')$

Before moving on to the analysis of presupposition in FCS, let's consider how we might analyse conjunction/discourse sequencing within this framework.

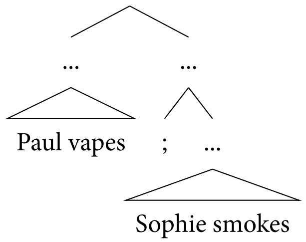
Consider the discourse in (7). Suppose that at c_1, the participants are ignorant about both whether Paul vapes and whether Sophie smokes. At c_2 the participants know that Paul vapes, but are ignorant about whether Sophie smokes. At c_3, the participants know both that Paul vapes and that Sophie smokes.

(7) c_1 Paul vapes c_2; Sophie smokes c_3.

We can give a *semantics* of the discourse sequencing operator (;) which captures the intuition that a sequential utterances successively update the context set. To sequence CCPs, we simply *compose* them.

(8) $(;) \coloneqq \lambda q \,.\, \lambda p \,.\, q \circ p$

(9) $\lambda c \,.\, [\mathbb{A}\ [\![\text{Sophie smokes}]\!]\,]\,([\mathbb{A}\ [\![\text{Paul vapes}]\!]\,]\ c)$



As it stands, the CCPs we've dealt with so far have been *total* functions over contexts. Furthermore, CCPs are defined in terms of conjunction, and since conjunction is associative:

(10) Paul vapes; Sophie smokes = Sophie smokes; Paul vapes.

In fact, if we lower the result of sequencing the two sentences, we just get...classical conjunction. So what have we achieved exactly?

Get them to demonstrate this in the exercises.

2.2 Presupposition satisfaction

Heim's (1983) innovation was to treat presuppositions as, essentially, *definedness conditions on CCPs*. We say that, if a context set c entails the presupposition of a sentence S, c **satisfies** the presupposition S.[1]

We can now shift gears and treat the CCPs associated with sentences as (potentiall!) partial functions from contexts to contexts. The CCP associated with the presuppositional sentence "Paul quit vaping" is given below:

(12) $\lambda c \,.\, \begin{cases} \lambda w \,.\, c\ w \wedge \neg\, \mathsf{p}\ \mathsf{vapes}_w & c \subseteq \{\, w \mid \mathsf{p}\ \mathsf{used\text{-}to\text{-}vape}_w \,\} \\ \sharp & \mathsf{else} \end{cases}$

Now that CCPs can be *partial*, treating discourse sequencing as function composition has an interesting consequence.

[1] You can also write definedness conditions on functions using Heim & Kratzer's colon notation, although it will quickly get quite unwieldy. For example:

(11) $\lambda c \,.\, c \subseteq \{\, w \mid \mathsf{p}\ \mathsf{used\text{-}to\text{-}vape}_w \,\} \,.\, \lambda w \,.\, c\ w \wedge \neg\, \mathsf{p}\ \mathsf{vapes}_w$

2.3 Quantification

2.3.1 Assignment functions and variables

In order to deal with anaphora, Heim's dynamic semantics deals with an enriched notion of context set – namely, a *file*. A file is a set of *world-assignment function pairs* (or equivalently, a relation between worlds and assignments).

The file associated with the information that a person at identifier 7 vapes is:

(13) $\{\, \langle g, w \rangle \ :\ g_7\ \mathsf{vapes}_w \,\}$

(14) Definition of a *file*:
c is a file iff there is a subset $\mathbb{N}'$ of $\mathbb{N}$, and c is a set of assignment-world pairs, $\langle g, w \rangle$, where $g : \mathbb{N}' \mapsto D_e$. We refer to $\mathbb{N}'$ as the *domain* of the file (i.e., $\mathsf{dom}\ c$)

We'll use variables named c, c', c''... to range over *files*. λc . ... can be read as an abbreviation of $\lambda c : \mathsf{file}\ c$ . ...

$g[n]g'$ mean that g and g' differ only at n, and that g_n is defined (g'_n may be undefined).

We can now give a CCP for a sentence with an indefinite:

(15) $\text{Someone}^7 \text{ arrived} = \lambda c \,.\, \{\, \langle g, w \rangle \mid \exists \langle g', w \rangle \in c[g[7]g' \wedge g_7\ \mathsf{arrived}_w] \,\}$

References

Heim, Irene. 1983. On the projection problem for presuppositions. In *Proceedings of WCCFL 2*, 114–125. Stanford University.
Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar* (Blackwell textbooks in linguistics 13). Malden, MA: Blackwell. 324 pp.