Introduction to File-Change Semantics

Chris Potts, Ling 230b: Advanced semantics and pragmatics, Spring 2018

May 30

1 Discourse referents

From McCawley's introduction to Karttunen's (1976) 'Discourse referents'

> The most interesting idea in Karttunen's paper appears not to have registered on many linguists or logicians. That is the idea that existential quantifiers have the dual function of asserting existence (thus binding a variable) and of introducing a constant that can figure in subsequent discourse. This idea is a vindication of the informal notational practise of mathematicians, who will write an existentially quantified formula [. . .] and thenceforth use the variable bound by the existential quantifier as if it were a constant [. . .]. Karttunen's observations provide a case for according this practice full status as part of a system of representing logical structure; that proposal allows one to cope with examples like *I have a proof of this theorem, but it won't fit in this margin*, which are a horror to accomodate [*sic*] within standard logical notation (there is no "standard" logical representation of it that has a constituent corresponding to *I have a proof of this theorem*, though there must be such a constituent, since the sentence can be continued *though Fermat says that he does and that it will fit in the margin of his copy of Euclid*).

2 A word from first-order logic

(1) $[\![\exists x(\textbf{sleepy}\ x)]\!]^{\mathbf{M},g} = \mathtt{T}$ iff **there exists an** assignment g' that differs from g at most on the value it assigns to x such that $g'(x)$ is a member of $[\![\textbf{sleepy}]\!]^{M,g'}$

Closed formulae denote either the full set of assignment functions or none of them. To see this, notice that it doesn't matter which assignment we begin with, because we are allowed to change its value for the bound variable. In contrast, formulae with free variables can denote any subset of the full set of assignments:

(2) $[\![(\textbf{sleepy}\ x)]\!]^{\mathbf{M}}$ = the set of all assignments g such that $g(x)$ is a member of $[\![\textbf{sleepy}]\!]^{\mathbf{M}}$

3 Three kinds of binding

3.1 Deixis

(3) The local semanticists hang out in Building 460.

(4) The coffee is strong!

3.2 Discourse anaphora

(5) We visited New York. The local semanticists took us out for beers.

(6) We visited Coupa Cafe. The coffee was strong.

3.3 Bound variable

(7) Every fan watched the game in a local bar.

(8) Every farmer who owns a donkey beats it.

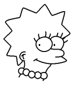
4 Central notions

- We are going to focus on capturing discourse binding.
- We are going to view things from the perspective of assignment functions.
- Basically all our formulae will contain free variables. Thus, basically all our formulae will denote sets of assignments (Heim calls them sequences).
- Indefinites extend the length of the domains of the assignments in our information state.
- Definites are defined only if the sequences in the information state already contain their indices.

5 Example model

5.1 Fixed world

$\llbracket \mathbf{l} \rrbracket =$



$\llbracket \mathbf{h} \rrbracket =$

$\llbracket \mathbf{b} \rrbracket =$

$$\llbracket \textbf{young} \rrbracket = \left\{ \llbracket \mathbf{l} \rrbracket, \llbracket \mathbf{b} \rrbracket \right\}$$

$$\llbracket \textbf{male} \rrbracket = \left\{ \llbracket \mathbf{h} \rrbracket, \llbracket \mathbf{b} \rrbracket \right\}$$

$$\llbracket \textbf{female} \rrbracket = \left\{ \llbracket \mathbf{l} \rrbracket \right\}$$

$$\llbracket \textbf{see} \rrbracket = \left\{ \begin{array}{c} \left\langle \llbracket \mathbf{l} \rrbracket, \llbracket \mathbf{l} \rrbracket \right\rangle, \\ \left\langle \llbracket \mathbf{b} \rrbracket, \llbracket \mathbf{l} \rrbracket \right\rangle, \\ \left\langle \llbracket \mathbf{h} \rrbracket, \llbracket \mathbf{l} \rrbracket \right\rangle, \end{array} \right\} \qquad \llbracket \textbf{tease} \rrbracket = \left\{ \begin{array}{c} \left\langle \llbracket \mathbf{h} \rrbracket, \llbracket \mathbf{b} \rrbracket \right\rangle, \\ \left\langle \llbracket \mathbf{h} \rrbracket, \llbracket \mathbf{l} \rrbracket \right\rangle, \\ \left\langle \llbracket \mathbf{h} \rrbracket, \llbracket \mathbf{h} \rrbracket \right\rangle, \end{array} \right\}$$

5.2 All assignments up to length 2

$$\boxed{\begin{array}{ccc}
\left[1 \mapsto \llbracket \mathbf{b} \rrbracket \right] & \left[1 \mapsto \llbracket \mathbf{h} \rrbracket \right] & \left[1 \mapsto \llbracket \mathbf{l} \rrbracket \right] \\
\left[2 \mapsto \llbracket \mathbf{b} \rrbracket \right] & \left[2 \mapsto \llbracket \mathbf{h} \rrbracket \right] & \left[2 \mapsto \llbracket \mathbf{l} \rrbracket \right] \\
\begin{bmatrix} 1 \mapsto \llbracket \mathbf{b} \rrbracket \\ 2 \mapsto \llbracket \mathbf{b} \rrbracket \end{bmatrix} & \begin{bmatrix} 1 \mapsto \llbracket \mathbf{b} \rrbracket \\ 2 \mapsto \llbracket \mathbf{l} \rrbracket \end{bmatrix} & \begin{bmatrix} 1 \mapsto \llbracket \mathbf{b} \rrbracket \\ 2 \mapsto \llbracket \mathbf{h} \rrbracket \end{bmatrix} \\
\begin{bmatrix} 1 \mapsto \llbracket \mathbf{l} \rrbracket \\ 2 \mapsto \llbracket \mathbf{b} \rrbracket \end{bmatrix} & \begin{bmatrix} 1 \mapsto \llbracket \mathbf{l} \rrbracket \\ 2 \mapsto \llbracket \mathbf{l} \rrbracket \end{bmatrix} & \begin{bmatrix} 1 \mapsto \llbracket \mathbf{l} \rrbracket \\ 2 \mapsto \llbracket \mathbf{h} \rrbracket \end{bmatrix} \\
\begin{bmatrix} 1 \mapsto \llbracket \mathbf{h} \rrbracket \\ 2 \mapsto \llbracket \mathbf{b} \rrbracket \end{bmatrix} & \begin{bmatrix} 1 \mapsto \llbracket \mathbf{h} \rrbracket \\ 2 \mapsto \llbracket \mathbf{l} \rrbracket \end{bmatrix} & \begin{bmatrix} 1 \mapsto \llbracket \mathbf{h} \rrbracket \\ 2 \mapsto \llbracket \mathbf{h} \rrbracket \end{bmatrix}
\end{array}}$$

6 Example: extension and elimination

$$\boxed{\phantom{\left[1 \mapsto \llbracket \mathbf{b} \rrbracket \right]}} + \llbracket \exists x (\textbf{young}\ x) \rrbracket = \boxed{\left[1 \mapsto \llbracket \mathbf{b} \rrbracket \right] \qquad \left[1 \mapsto \llbracket \mathbf{l} \rrbracket \right]} + \llbracket (\textbf{female}\ x_1) \rrbracket =$$

$$\boxed{\left[1 \mapsto \llbracket \mathbf{l} \rrbracket \right]} + \llbracket \exists x (\textbf{see}\ x_1\ x) \rrbracket = \boxed{\begin{bmatrix} 1 \mapsto \llbracket \mathbf{l} \rrbracket \\ 2 \mapsto \llbracket \mathbf{b} \rrbracket \end{bmatrix} \begin{bmatrix} 1 \mapsto \llbracket \mathbf{l} \rrbracket \\ 2 \mapsto \llbracket \mathbf{l} \rrbracket \end{bmatrix} \begin{bmatrix} 1 \mapsto \llbracket \mathbf{l} \rrbracket \\ 2 \mapsto \llbracket \mathbf{h} \rrbracket \end{bmatrix}}$$

7 Definitions

Definition 1 (Domains)**.** Dom(F) = the set of all indices in F

Definition 2 (Satisfaction sets)**.** The satisfaction set for a file F, written Sat(F), is the set of all sequences that satisfy all the formulae in F.

Definition 3 (Updates)**.** To compute Sat($F + p$) (as defined in Heim 1983:(18)), gather together all the assignments in Sat(F) and all the assignments that satisfy p. Now consider the union. This must also be a function, which means that the two satisfaction sets must agree on all mappings that they share. The result is that the sequences in Sat(F) are 'extended' by those that satisfy p.

Definition 4 (Novelty and familiarity; Heim 1983:(15))**.** Let F be a file, p an atomic proposition. Then p is appropriate with respect to F only if, for every noun phrase NP$_{ß}$ with index i that p contains

- if NP$_i$ is definite, then $i \in$ Dom(F)
- and if NP$_i$ is indefinite, then $i \notin$ Dom(F)

Definition 5 (Universal quantification; Heim 1983:(34))**.** To update a file card F with $\forall x((\varphi\ x) \rightarrow (\psi\ x))$,

i. Update F with $\exists(\varphi\ x)$ to obtain F':

ii. Update F' with $(\psi\ x)$, where x is now definite, to form F''.

iii. Check to ensure that for every sequence in F that was extended in F' is also in F''.

Definition 6 (Negation ; Heim 1983:(35))**.** Like universal quantification, negation is a 'test'. For Sat($F + \neg p$), if there is no sequence in Sat(F) that can be extended to make p true, then the result of the update is Sat(F). If there is such a sequence in Sat(F), then the result of the update is the empty set.

8 Examples to complete

8.1 Extended sequence

$+[\![\exists x(\textbf{see lisa } x)]\!] =$

$+[\![(\textbf{male } x_1)]\!] =$

$+[\![\exists x(\textbf{tease } x \; x_1)]\!] =$

8.2 Double indefinite

$+[\![\exists x \exists y(\textbf{see } y \; x)]\!] =$

8.3 Simple universal quantifier

$$\boxed{\left[1 \mapsto [\![\mathbf{h}]\!]\right]} + [\![\forall x((\mathbf{young}\ x) \to (\mathbf{male}\ x))]\!] = \boxed{}$$

Auxiliary boxes: the left extends every sequence in the input with a fresh index satisfying the restriction. The second restricts to the subset that satisfy the consequent. Since the left is a not a subset of the right, the formula is false, hence the empty output above.

$$\boxed{\begin{bmatrix}1 \mapsto [\![\mathbf{h}]\!] \\ 2 \mapsto [\![\mathbf{b}]\!]\end{bmatrix} \begin{bmatrix}1 \mapsto [\![\mathbf{h}]\!] \\ 2 \mapsto [\![\mathbf{l}]\!]\end{bmatrix}} \quad \boxed{\begin{bmatrix}1 \mapsto [\![\mathbf{h}]\!] \\ 2 \mapsto [\![\mathbf{b}]\!]\end{bmatrix}}$$

Compare $\forall x((\mathbf{young}\ x) \to (\mathbf{tease}\ x\ \mathbf{h})))$. This is true in our model. In that case, the statement leaves the original satisfaction set unchanged.

8.4 A 'donkey sentence': *Every male who teased a female saw her*

$$\forall x\Big(\big((\mathbf{male}\ x) \wedge \exists y(\mathbf{female}\ y) \wedge \mathbf{tease}\ y\ x)\big) \to \big(\mathbf{see}\ y\ x\big)\Big)$$

Auxiliary boxes (assuming an empty input context):

$$\exists x_1(\mathbf{male}\ x_1) \wedge \exists x_2\big((\mathbf{female}\ x_2)(\mathbf{tease}\ x_2\ x_1)\big)\Big) \qquad (\mathbf{see}\ x_2\ x_1)$$

$$\boxed{\begin{bmatrix}1 \mapsto [\![\mathbf{h}]\!] \\ 2 \mapsto [\![\mathbf{l}]\!]\end{bmatrix}} \qquad \boxed{\begin{bmatrix}1 \mapsto [\![\mathbf{h}]\!] \\ 2 \mapsto [\![\mathbf{l}]\!]\end{bmatrix}}$$

8.5 Universal with indefinite in its nuclear scope

$$\forall x\Big(\big(\mathbf{male}\ x\big) \rightarrow \big(\exists y\ (\mathbf{see}\ y\ x)\big)\Big)$$

Auxiliary boxes (assuming an empty input context):

$$\boxed{\big[1 \mapsto \llbracket \mathbf{b} \rrbracket\big]\ \big[1 \mapsto \llbracket \mathbf{h} \rrbracket\big]}\quad \boxed{\begin{bmatrix} 1 \mapsto \llbracket \mathbf{b} \rrbracket \\ 2 \mapsto \llbracket \mathbf{l} \rrbracket \end{bmatrix} \quad \begin{bmatrix} 1 \mapsto \llbracket \mathbf{h} \rrbracket \\ 2 \mapsto \llbracket \mathbf{l} \rrbracket \end{bmatrix}}$$

So this is true because every sequence in the left box has a (proper) extension in the right box.

8.6 Negated indefinites

$$\boxed{\big[1 \mapsto \llbracket \mathbf{h} \rrbracket\big]} + \llbracket \neg\exists x(\mathbf{young}\ x) \rrbracket = \boxed{\quad}$$

Auxiliary box: since this is not empty, the update returns the empty set.

$$\boxed{\begin{bmatrix} 1 \mapsto \llbracket \mathbf{h} \rrbracket \\ 2 \mapsto \llbracket \mathbf{b} \rrbracket \end{bmatrix} \begin{bmatrix} 1 \mapsto \llbracket \mathbf{h} \rrbracket \\ 2 \mapsto \llbracket \mathbf{l} \rrbracket \end{bmatrix}}$$

Thus, a single negation shuts down the dynamics of its complement. A second negation does not restore those dynamics, though. Consider $\neg\neg\exists x(\mathbf{young}\ x)$. Since we saw that the singly negated form returns the empty set, the negation of that just brings us back to our original input state. No discourse referent is created by the existential.

References and closely related work

Beaver, David I. 2001. *Presupposition and assertion in dynamic semantics*. Stanford, CA: CSLI.

Chierchia, Gennaro. 1995. *Dynamics of meaning: Anaphora, presupposition, and the theory of grammar*. Chicago: University of Chicago Press.

Dekker, Paul. 1994. Predicate logic with anaphora. In Lynn Santelmann & Mandy Harvey (eds.), *Proceedings from SALT IX*, 79–95. Ithaca, NY: DMLL Publications, Cornell University.

Groenendijk, Jeroen & Martin Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy* 14(1). 39–100.

Heim, Irene. 1982. *The semantics of definite and indefinite noun phrases*: UMass Amherst dissertation.

Heim, Irene. 1983. On the projection problem for presuppositions. In Michael Barlow, Daniel P. Flickinger & Michael T. Wescoat (eds.), *Proceedings of the 2nd west coast conference on formal linguistics*, 114–125. Stanford, CA: Stanford Linguistics Association.

Kamp, Hans & Uwe Reyle. 1993. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Dordrecht: Kluwer.

Karttunen, Lauri. 1976. Discourse referents. In James D. McCawley (ed.), *Syntax and semantics*, vol. 7: Notes from the Linguistic Underground, 363–385. New York: Academic Press.