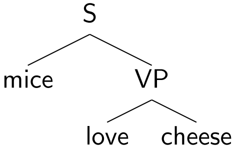
Inner/Outer Meanings in Formal Semantics perspective

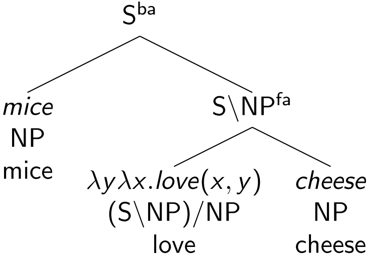
Phong Le

February 5, 2014

Example: Mice love cheese



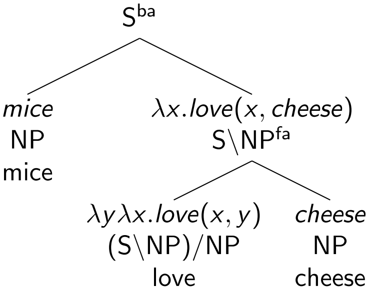
Formal semantics perspective



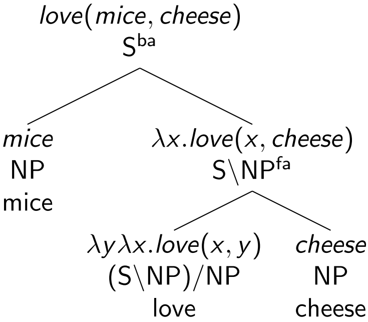
Five CCG rules

- Forward Application (fa)
  $X/Y : P \quad Y : Q \Rightarrow X : PQ$
- Backward Application (ba)
  $Y : P \quad X\backslash Y : Q \Rightarrow X : QP$
- Composition (comp)
  $X/Y : P \quad Y/Z : Q \Rightarrow X/Z : \lambda x.(P(Qx))$
- Coordination (conj)
  $X : P \;\; conj \;\; X' : P' \Rightarrow X'' : \lambda x.(Px \wedge P'x)$
- Type raising (tr)
  $NP : a \Rightarrow T/(T\backslash NP) : \lambda R.(Ra)$

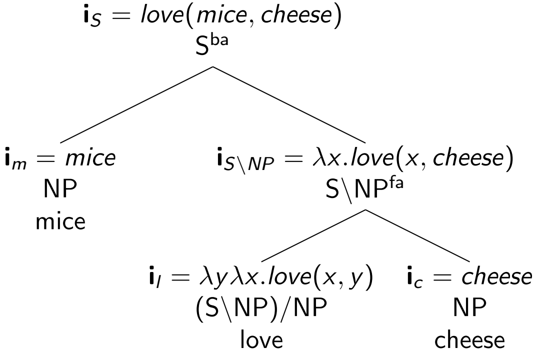
Formal semantics perspective



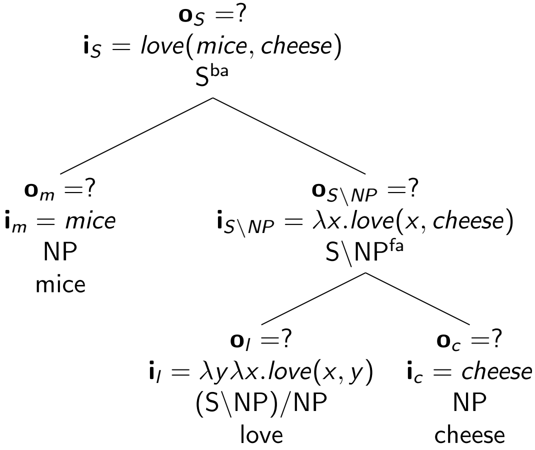
Formal semantics perspective (cont.)



They are "inner" meanings



How about “outer” meanings?



Key point

Outer meanings (i.e., context representations) are for word/phrase prediction.

Example

World model M:

- Mice love cheese
- Tom love cheese
- Mice hate cats

What can we predict about X given a context:

- X love cheese $\rightarrow X \in \{mice, Tom\}$
- mice X cheese $\rightarrow X \in \{love\}$

Proposal

$M \models \mathbf{o}(\mathbf{i})$

- Define "variable-swapping" function: $\gamma(\lambda x \lambda y.Q) = \lambda y \lambda x.Q$ ($\gamma = \lambda U.\lambda y \lambda x.Uxy$)
- Define six "variable-raising" functions

$$
\begin{aligned}
\beta_1(\lambda P.P) &= \lambda x \lambda P.Px \\
\beta_2(\lambda P.Pv) &= \lambda y \lambda P.Pyv \\
\beta_3(\lambda P.P) &= \lambda Q \lambda P.(QP) \\
\beta_4(\lambda P.Pv) &= \lambda Q \lambda P.(Q(Pv)) \\
\beta_5(\lambda P.P) &= \lambda Q \lambda P.(Q \wedge P) \\
\beta_6(\lambda P.Pv) &= \lambda Q \lambda P.(Qv \wedge Pv)
\end{aligned}
$$

Five extended CCG rules

- Forward Application (fa)

$$\begin{array}{c} \mathbf{o}_X \\ \mathbf{i}_X = PQ \end{array}$$

$$\begin{array}{cc} \mathbf{o}_{X/Y} = \beta_k(\mathbf{o}_X)\mathbf{i}_Y & \mathbf{o}_Y = \gamma(\beta_k(\mathbf{o}_X))\mathbf{i}_{X/Y} \\ \mathbf{i}_{X/Y} = P & \mathbf{i}_Y = Q \end{array}$$

($k = 2$ if X is a complex type, otherwise $k = 1$)

- Backward Application (ba)

$$\begin{array}{c} \mathbf{o}_X \\ \mathbf{i}_X = QP \end{array}$$

$$\begin{array}{cc} \mathbf{o}_Y = \gamma(\beta_k(\mathbf{o}_X))\mathbf{i}_{X\backslash Y} & \mathbf{o}_{X\backslash Y} = \beta_k(\mathbf{o}_X)\mathbf{i}_Y \\ \mathbf{i}_Y = P & \mathbf{i}_{X\backslash Y} = Q \end{array}$$

($k = 2$ if X is a complex type, otherwise $k = 1$)

- Composition (comp)

$$\mathbf{o}_{X/Z}$$
$$\mathbf{i}_{X/Z} = \lambda y.(P(Qy))$$

$$\mathbf{o}_{X/Y} = \gamma(\beta_4(\mathbf{o}_{X/Z}))\mathbf{i}_{Y/Z} \qquad \mathbf{o}_{Y/Z} = \beta_4(\mathbf{o}_{X/Z}))\mathbf{i}_{X/Y}$$
$$\mathbf{i}_{X/Y} = P \qquad \mathbf{i}_{Y/Z} = Q$$

- Coordination (conj)

$$\mathbf{o}_{X''}$$
$$\mathbf{i}_{X''} = \lambda x.(Qx \wedge Px)$$

$$\mathbf{o}_{X} = \gamma(\beta_k(\mathbf{o}_{X''}))\mathbf{i}_{X'} \qquad \mathbf{o}_{X'} = \beta_k(\mathbf{o}_{X''})\mathbf{i}_{X}$$
$$\mathbf{i}_{X} = P \qquad \mathbf{i}_{X'} = P'$$

($k = 6$ if X'' is a complex type, otherwise $k = 5$)

- Type raising (tr)

$$\begin{array}{c} \mathbf{o}_{T/(T\backslash NP)} \\ \mathbf{i}_{T/(T\backslash NP)} = \lambda R.(Ra) \\ | \\ \mathbf{o}_{NP} = \mathbf{o}_{T/(T\backslash NP)}(\lambda R.R) \\ \mathbf{i}_{NP} = a \end{array}$$

Example: Mice love cheese

Compute $\mathbf{o}_S$

$$\boxed{\mathbf{o}_S = \lambda P.P}$$
$$\mathbf{i}_S = \mathit{love}(\mathit{mice}, \mathit{cheese})$$

$\mathbf{o}_m = ?$
$\mathbf{i}_m = \mathit{mice}$
mice

$\mathbf{o}_{S\backslash NP} = ?$
$\mathbf{i}_{S\backslash NP} = \lambda x.\mathit{love}(x, \mathit{cheese})$

$\mathbf{o}_l = ?$
$\mathbf{i}_l = \lambda y \lambda x.\mathit{love}(x, y)$
love

$\mathbf{o}_c = ?$
$\mathbf{i}_c = \mathit{cheese}$
cheese

$\mathbf{o}_S(\mathbf{i}_S) = (\lambda P.P)(\mathit{love}(\mathit{mice}, \mathit{cheese})) = \mathit{love}(\mathit{mice}, \mathit{cheese})$
Hence, $M \models \mathbf{o}_S(\mathbf{i}_S)$

Compute $\mathbf{o}_m$

$$\boxed{\mathbf{o}_S = \lambda P.P}$$
$$\mathbf{i}_S = love(mice, cheese)$$

$$\boxed{\mathbf{o}_m = ?}$$
$$\mathbf{i}_m = mice$$
mice

$$\mathbf{o}_{S\backslash NP} = ?$$
$$\boxed{\mathbf{i}_{S\backslash NP} = \lambda x.love(x, cheese)}$$

$$\mathbf{o}_l = ?$$
$$\mathbf{i}_l = \lambda y \lambda x.love(x, y)$$
love

$$\mathbf{o}_c = ?$$
$$\mathbf{i}_c = cheese$$
cheese

$$\begin{aligned}\mathbf{o}_m &= comp_{ba,left}(\mathbf{o}_S, \mathbf{i}_{S\backslash NP}) = \gamma(\beta_1(\mathbf{o}_S))\mathbf{i}_{S\backslash NP} \\ &= (\lambda P \lambda x.Px)(\lambda x.love(x, cheese)) = \lambda x.love(x, cheese)\end{aligned}$$

$\mathbf{o}_m(\mathbf{i}_m) = \big(\lambda x.love(x, cheese)\big)(mice) = love(mice, cheese)$
Hence, $M \models \mathbf{o}_m(\mathbf{i}_m)$

Compute $\mathbf{o}_{S\backslash NP}$

$$\boxed{\mathbf{o}_S = \lambda P.P}$$
$$\mathbf{i}_S = love(mice, cheese)$$

$$\mathbf{o}_m = ?$$
$$\boxed{\mathbf{i}_m = mice}$$
mice

$$\boxed{\mathbf{o}_{S\backslash NP} = ?}$$
$$\mathbf{i}_{S\backslash NP} = \lambda x.love(x, cheese)$$

$$\mathbf{o}_l = ?$$
$$\mathbf{i}_l = \lambda y \lambda x.love(x, y)$$
love

$$\mathbf{o}_c = ?$$
$$\mathbf{i}_c = cheese$$
cheese

$$\begin{aligned}\mathbf{o}_{S\backslash NP} &= comp_{ba,right}(\mathbf{o}_S, \mathbf{i}_m) = \beta_1(\mathbf{o}_S)(\mathbf{i}_m) \\ &= \big(\lambda x \lambda P.Px\big)\big(mice\big) = \lambda P.P\ mice\end{aligned}$$

$\mathbf{o}_{S\backslash NP}(\mathbf{i}_{S\backslash NP}) = \big(\lambda P.P\ mice\big)(\lambda x.love(x, cheese)) = love(mice, cheese)$
Hence, $M \models \mathbf{o}_{S\backslash NP}(\mathbf{i}_{S\backslash NP})$

Compute $\mathbf{o}_l$

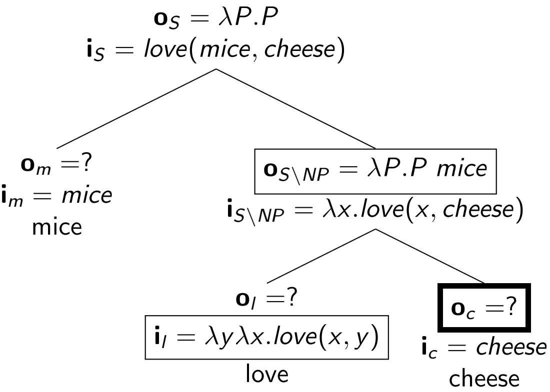
$$\mathbf{o}_S = \lambda P.P$$
$$\mathbf{i}_S = love(mice, cheese)$$

- $\mathbf{o}_m =?$
$\mathbf{i}_m = mice$
mice
- $\boxed{\mathbf{o}_{S\backslash NP} = \lambda P.P\ mice}$
$\mathbf{i}_{S\backslash NP} = \lambda x.love(x, cheese)$
  - $\boxed{\mathbf{o}_l =?}$
$\mathbf{i}_l = \lambda y \lambda x.love(x, y)$
love
  - $\mathbf{o}_c =?$
$\boxed{\mathbf{i}_c = cheese}$
cheese

$$\begin{aligned}\mathbf{o}_l &= comp_{fa,left}(\mathbf{o}_{S\backslash NP}, \mathbf{i}_c) = \beta_2(\mathbf{o}_{S\backslash NP})(\mathbf{i}_c) \\ &= \big(\lambda y \lambda P.P\ y\ mice\big)\big(cheese\big) = \lambda P.P\ cheese\ mice\end{aligned}$$

$\mathbf{o}_l(\mathbf{i}_l) = \big(\lambda P.P\ cheese\ mice\big)(\lambda y \lambda x.love(x, y)) = love(mice, cheese)$
Hence, $M \models \mathbf{o}_l(\mathbf{i}_l)$

Compute $\mathbf{o}_c$



$$\begin{aligned}\mathbf{o}_c &= comp_{fa,right}(\mathbf{o}_{S\backslash NP}, \mathbf{i}_l) = \gamma(\beta_2(\mathbf{o}_{S\backslash NP}))(\mathbf{i}_l) \\ &= \big(\lambda P \lambda y.P\ y\ mice\big)\big(\lambda y \lambda x.love(x, y)\big) = \lambda y.love(mice, y)\end{aligned}$$

$\mathbf{o}_c(\mathbf{i}_c) = (\lambda y.love(mice, y))(cheese) = love(mice, cheese)$
Hence, $M \models \mathbf{o}_l(\mathbf{i}_l)$

Formal semantics perspective vs IORNN

	Formal semantics	IORNN
$\mathbf{o}_S$	$= \lambda P.P$	$= \mathbf{o}_\emptyset$
Compose inner meanings	five CCG rules	$\mathbf{i}_{p_1} = f(\mathbf{W}^i_1 \mathbf{i}_x + \mathbf{W}^i_2 \mathbf{i}_y + \mathbf{b}^i)$
Compose outer meanings	five extended CCG rules	$\mathbf{o}_{p_1} = g(\mathbf{W}^o_1 \mathbf{o}_{p_2} + \mathbf{W}^o_2 \mathbf{i}_z + \mathbf{b}^o)$
Criterion	$M \models \mathbf{o}(\mathbf{i})$ (binary scores)	correct words/phrases are given higher scores than others (continuous scores)
Outer meanings' role	select words/phrases that make complete sentences correct in a given world model	select words/phrases that tend to occur in given contexts