CSC2519F: Natural Language Semantics, Lecture 4

Proof theory for HOL

(1) $\vdash P(\top)\&P(\bot) = \forall x P(x)$.

(2) $\vdash (x = y) \rightarrow (P(x) = P(y))$.

(3) $\vdash (x = y) \rightarrow \forall z(x(z) = y(z))$.

(4) $\vdash (\lambda x.\alpha)\beta = \alpha[x \mapsto \beta$, where β is free for x in α.

(5) $\vdash \iota(eq_e(y)) = y$.

In addition to these axioms ther is a rule of inference which is as follows.

$$\frac{\phi[a/x] \quad \alpha = \beta}{\phi[\beta/x]}\ \text{subst}$$

This proof system is sound and complete.

ACG derivations

Let $G = \langle BasCat, \tau, s\rangle$ be an ACG, where

- $BasCat \subseteq Cat$,
- $\tau : \Sigma \rightarrow P(Cat)$,
- $s \in BasCat$.

$$\mathcal{L}(G) = \{w \in \Sigma^* \mid \exists\Gamma \in \tau^*(w) \text{ such that } \Gamma \rightarrow s\}$$

There is more than one way to interpret τ.

- "Saussurrean": $\tau : Cat \rightarrow \Sigma \times Term$ (where Σ is like a por representation of the sound of words, and $Term$ is the meaning of words).

- "Categorical grammar": $\tau : \Sigma \rightarrow Cat \times Term$. E.g.:
  john ⇒ np : john
  the ⇒ np /n: ι_e
  boy ⇒ n : boy
  tall ⇒ n /n : tall
  loves ⇒ np \s /np : loves

Why are terms the same as Σ-expressions? There are two reasons:

1. β-reduction. E.g.:
   boy ⇒ n : boy
   loves ⇒ np \s /np : loves $\quad \equiv \lambda x.\lambda y.loves(x)(y)$
2. "Meaning postulates". E.g.:
   tall ⇒ n /n : tall $\quad \equiv \lambda P.\lambda x.tall(x)\&P(x)$

Transformations between types:

- e to $e \rightarrow t$, using $Q \equiv \lambda y.(x = y)$.
- $e \rightarrow t$ to e, using ι.
- $e \rightarrow t$ to $(e \rightarrow t) \rightarrow t$, using some $\equiv \lambda P.\lambda Q.(Q \cap P) = \emptyset$.
- e to $(e \rightarrow t) \rightarrow t$, using R, where $R \not\equiv \lambda P.P(x)$, but $R \equiv \lambda P.\lambda Q.\lambda x.Q(x)\&P(x)$.

Another way to interpret "the":
the $\equiv \lambda P.\lambda Q.some(P)(Q)\&def(P)$
def $\equiv \lambda P.some(\lambda y.every(\lambda x.P(x) \Leftrightarrow x = 1))$ [Russel 05]

Some more examples:
outside ⇒ (np \s) \np \s : outside
in ⇒ (n \n) /np : in (Adjectival PP. E.g. the man in the blue suit.)
in ⇒ (np \s) \np \s /np : in' (Adverbial PP. E.g. Fido plays in the yard.)

Det ⇒ np /n : E.g. the, every , a
Prenom. adj. ⇒ n /n : E.g. tall
Postnom. adj. ⇒ n \n : E.g. alleged
Adj. P ⇒ (n \n) /np : E.g. in

Adv. P ⇒ ⇒ (np \s) \np \s /np : E.g. in
Preverbal VP-mod ⇒ (n \s) /(np \s) : E.g. barely
Trans verb ⇒ (np \s) /np : E.g. sees, loves
Intrans verb ⇒ np \s : E.g. walks
Ditrans verb ⇒ (np \s) /np /np : E.g. give
Coord. particles ⇒ X \X /X : E.g. and, or

In the Applicative Categorical calculus we have two basic rules:

$$\frac{A/B \quad B}{A}\ /e$$

$$\frac{B \quad B\backslash A}{A}\ \backslash e$$

Note that there is some variation in notation. $B\backslash A$ in the second rule is sometimes written as A/B. This is Steedman notation. Also, another way you can see the rules written is with a derivation $\Delta_1 \cdots \Delta_n$ on top of each of the A, B, A/B, $B\backslash A$.

We now extend these as follows in order to account for the interpretation of τ, $\tau : \Sigma \rightarrow Cat \times Term$.

$$\frac{A/B : \alpha \quad B : \beta}{A : \alpha(\beta)}\ /e$$

$$\frac{B : \beta \quad B\backslash A : \alpha}{A : \alpha(\beta)}\ \backslash e$$

This is called Labelled Term Deduction System and is considered an alternative to the Model-theoretic Semantics. Note also that the label $\alpha(\beta)$ is the same for notations $B\backslash A$.

Example derivation 1

$$\cfrac{\begin{array}{c}\text{John}\\ \text{np : john}\end{array} \quad \cfrac{\begin{array}{c}\text{sees}\\ \text{(np \textbackslash s) /np : sees} \equiv \lambda x.\lambda y.\text{sees}(y)(x)\end{array} \quad \begin{array}{c}\text{Mary}\\ \text{np : mary}\end{array}}{\text{np \textbackslash s : sees(mary)}}\ \text{/e}}{\text{s : sees(mary)(john)}}\ \text{\textbackslash e}$$

Note that in these derivations order and multiplicity matters and there should be no spare parts.

Example derivation 2

red	car	in	Richmond
n /n : red	n : car $\equiv \lambda x.\text{car}(x)$	(n \n) /np : in	np : R

———————————————— /e

n \n : in(R) $\equiv \lambda P.\lambda x.\text{in(R)}(x)\&P(x)$

———————————————— \e

n : in(R)(car)

———————————————— /e

n : red(in(R)(car))

$\equiv \lambda x.\text{in(R)}(\text{car})(x)\&\text{red}(x)$

$\equiv \lambda x.[\lambda y.\text{in(R)}(y)\&\text{car}(y)](x)\&\text{red}(x)$

The other derivation results in n: in(R)(red(car))
$\equiv \lambda x.\text{in(R)}(x)\&\text{red}(\text{car})(x)$
$\equiv \lambda x.\text{in(R)}(x)\&[\lambda y.\text{car}(y)\&\text{red}(y)](x)$

Example derivation 3

pyramid	on	the	box	near	the	table
n : pyr	(n \n) /np : on	np /n :the	n: box	(n \n) /np : near	np /n :the	n : table

———————— /e

np : the(table)

———————— /e

n \n : near(the(table))

Now here are two possible derivations. This is the first one:

————————

n

————————

np

————————

n \n

————————

n : on(the(near(the(table))(box)))(pyr)

This is the second one:

————————

np

————————

n \n

————————

n

————————

n : near(the(table))(on(the(box))(pyr))

Example derivation 4

picture	of	Gerald
n /np_{of} : picture	np_{of} /np_{obj} : $\lambda x.x$	np_{obj} : g

———————— /e

np_{of} : g

———————— /e

n : λy.picture$(g)(y)$

Here we followed the convention of the Generalized Phrase Structural Grammars (GPSG) and annotated types with cases.