Yasu Sudo
Autumn 2018

1 Compositional Semantics

- In PLIN2001 Semantic Theory, we developed a **compositional semantics** for (a fragment of) English. The lecture notes are available here: https://www.ucl.ac.uk/~ucjtudo/ST.html.
- We focus on **truth-conditional meaning** (meaning pertinent to truth-conditions).
- In order to account for the fact that native speakers have truth-conditional intuitions about infinitely many grammatical sentences, we assume that natural languages obey the **(Local) Compositionality Principle**.

> **The (Local) Compositionality Principle**
> The meaning of a syntactically complex phrase *A* is determined solely by the meaning of its immediate daughter constituents.

  ('Meaning' here means truth-conditional meaning)

- We adopt **Frege's Conjecture** and assume that the primary mode of semantic composition is **function application**: when two meanings combine, one is a function and the other one is its argument. The resulting meaning is the value the function returns for that argument.

2 Model Theory

- We formalize the above idea in **model-theoretic semantics**.
- Every grammatical expression (a single word or a phrase) in English is assigned a model-theoretic object as its **denotation** relative to a **model**.
  - A model *M* is a mathematical structure that models a particular state of affairs.
  - Formally, *M* is a pair $\langle D, I\rangle$, where *D* is a non-empty set of individuals/entities and *I* is a function that assigns meanings to constants.
  - We write $[\![\alpha]\!]^M$ to mean the denotation of a (grammatical) expression α relative to model *M*.
- Every denotation is one of three kinds:
  - Individuals/Entities
  - Truth-values (0 or 1)
  - Functions
- We use **semantic types** to label different kinds of semantic objects:

(1) *Semantic types*
  a. *e* is a type.
  b. *t* is a type.
  c. If σ and τ are both types, $\langle\sigma, \tau\rangle$ is also a type.
  d. Nothing else is a type.

(2) *Domains*
  a. D_e is the set of individuals, *D*.
  b. D_t is the set of truth-values, $\{0, 1\}$.

c. $D_{\langle \sigma, \tau \rangle}$ is the set of functions whose domain is D_σ and whose range is D_τ.

(We sometimes write *et* for $\langle e, t \rangle$)

- Our semantic theory has two components:
  - **Lexicon**: list of denotations of syntactically simple expressions (alt.: atomic expressions)
  - **Compositional Rules**: rules for how to combine meanings

3 Lexicon

In PLIN2001 Semantic Theory, we covered the following items.

- Proper names denote entities (which are of type e and members of D).
- Sentences denote truth-values (which are of type t and either 0 or 1).
- Intransitive predicates denote functions of type $\langle e, t \rangle$.

(3) For any model M,

a. $[\![\text{smokes}]\!]^M = [\lambda x \in D_e.\ 1 \text{ iff } x \text{ smokes in } M]$

b. $[\![\text{linguist}]\!]^M = [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is a linguist in } M]$

c. $[\![\text{British}]\!]^M = [\lambda x \in D_e.\ 1 \text{ iff } x \text{ is British in } M]$

- Transitive predicates denote functions of type $\langle e, \langle e, t \rangle\rangle$.

(4) For any model M,

a. $[\![\text{likes}]\!]^M = [\lambda x \in D_e.[\lambda y \in D_e.\ 1 \text{ iff } y \text{ likes } x \text{ in } M]]$

b. $[\![\text{part}]\!]^M = [\lambda x \in D_e.[\lambda y \in D_e.\ 1 \text{ iff } y \text{ is part of } x \text{ in } M]]$

c. $[\![\text{fond}]\!]^M = [\lambda x \in D_e.[\lambda y \in D_e.\ 1 \text{ iff } y \text{ is fond of } x \text{ in } M]]$

d. $[\![\text{from}]\!]^M = [\lambda x \in D_e.[\lambda y \in D_e.\ 1 \text{ iff } y \text{ is from } x \text{ in } M]]$

- Semantically vacuous items denote identity functions.

(5) For any model M,

a. $[\![\text{a}_{\text{predicative}}]\!]^M = [\lambda P \in D_{\langle e,t \rangle}.\ P]$

b. $[\![\text{is}]\!]^M = [\lambda P \in D_{\langle e,t \rangle}.\ P]$

c. $[\![\text{of}]\!]^M = [\lambda x \in D_e.\ x]$

- If you do not remember what functions are or how the lambda notation for functions works, read the following lecture notes:
  - Functions
  - More on functions
  - Lambda notation

4 Compositional Rules

- These meanings combine via various **compositional rules**.
- For branching structures, we have three compositional rules:

Functional Application (FA)
For any model M, if A is a branching node whose immediate daughter constituents are B and C such that $[\![\text{C}]\!]^M \in \text{dom}([\![\text{B}]\!]^M)$, then $[\![\text{A}]\!]^M = [\![\text{B}]\!]^M([\![\text{C}]\!]^M)$.

Predicate Modification (PM)
For any model M, if A is a branching node with children B and C such that $\llbracket \text{B} \rrbracket^M$ and $\llbracket \text{C} \rrbracket^M$ are both of type $\langle e, t\rangle$, then $\llbracket \text{A} \rrbracket^M = [\lambda x \in D_e.\ \llbracket \text{B} \rrbracket^M(x) = \llbracket \text{C} \rrbracket^M(x) = 1]$.

Non-Branching Node Rule (NB)
For any model M, $\left\llbracket \begin{array}{c} \text{A} \\ | \\ \text{B} \end{array} \right\rrbracket^M = \llbracket \text{B} \rrbracket^M$.

- E.g. suppose $\llbracket \text{Cate} \rrbracket^{M_2} = c$ and $\llbracket \text{Denis} \rrbracket^{M_2} = d$:

$$
\begin{aligned}
\left\llbracket \begin{array}{c} \text{S} \\ \text{Cate} \quad \begin{array}{c} \text{VP} \\ \text{saw} \quad \text{Denis} \end{array} \end{array} \right\rrbracket^{M_2}
&= \left\llbracket \begin{array}{c} \text{VP} \\ \text{saw} \quad \text{Denis} \end{array} \right\rrbracket^{M_2} \left(\llbracket \text{Cate} \rrbracket^{M_2} \right) && \text{(BNR)} \\
&= \left\llbracket \begin{array}{c} \text{VP} \\ \text{saw} \quad \text{Denis} \end{array} \right\rrbracket^{M_2} (c) && \text{(BNR)} \\
&= \llbracket \text{saw} \rrbracket^{M_2}(\llbracket \text{Denis} \rrbracket^{M_2})(c) \\
&= \llbracket \text{saw} \rrbracket^{M_2}(d)(c) \\
&= [\lambda x \in D_e.[\lambda y \in D_e.\ 1 \text{ iff } y \text{ saw } x \text{ in } M_2]](d)(c) \\
&= [\lambda y \in D_e.\ 1 \text{ iff } y \text{ saw } d \text{ in } M_2](c) && (\lambda\text{-conv.}) \\
&= 1 \text{ iff } c \text{ saw } d \text{ in } M_2 && (\lambda\text{-conv.})
\end{aligned}
$$