1 Review

1.1 Set theory

1.2 Propositional logic

1.3 Predicate logic

- Syntax of predicate logic
- Semantics of predicate logic
  - Interpretation functions and models
    * A model $M$ is a pair $\langle D, I \rangle$.
    * $D$ is the domain, a (nonempty) set of individuals.
    * $I$ is an interpretation function: an assignment of semantic values to every basic expression (constant) in the language.
  - Assignment functions ($g$) assigns individuals in $D$ to individual variables in formulas.
  - Properties of relations
    (1) a. Symmetry: $\forall x \forall y (R(x, y) \rightarrow R(y, x))$
    b. Nonsymmetry: $\neg \forall x \forall y (R(x, y) \rightarrow R(y, x))$
    c. Asymmetry: $\forall x \forall y (R(x, y) \rightarrow \neg R(y, x))$

1.4 Type theory

- Fregean Principle: The meaning of a complex expression should be a function of the meaning of its parts.
  - unsaturated parts vs. saturated parts
  - functional application (FA)
- Categorical grammar
  - Basic categories: S, N
- Derived categories: S/S, S/SS, S/N, ...

- Semantic types
  - basic types: e, t
  - complex types:
    1. $e \in T, t \in T$
    2. If $\alpha \in T, \beta \in T$, then $<\alpha, \beta> \in T$
    3. Nothing is an element of $T$ except on the basis of (a), (b), and (c).

<table>
<thead>
<tr>
<th>Syntactic category</th>
<th>Semantic type</th>
<th>Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>t</td>
<td>sentences</td>
</tr>
<tr>
<td>ProperN</td>
<td>e</td>
<td>names (John)</td>
</tr>
<tr>
<td>CN(P)</td>
<td>e \rightarrow t</td>
<td>common noun phrases (cat)</td>
</tr>
<tr>
<td>NP</td>
<td>(i) e</td>
<td>“e-type” or “referential” NPs (John, the king)</td>
</tr>
<tr>
<td></td>
<td>(ii) e \rightarrow t</td>
<td>NPs as predicates (a student, a king)</td>
</tr>
<tr>
<td>ADJ(P)</td>
<td>e \rightarrow t</td>
<td>predicative adjectives (carnivorous, happy)</td>
</tr>
<tr>
<td>REL</td>
<td>e \rightarrow t</td>
<td>relative clauses (who(m) Mary loves)</td>
</tr>
<tr>
<td>VP, IV</td>
<td>e \rightarrow t</td>
<td>verb phrases, intransitive verbs (loves Mary, is tall)</td>
</tr>
<tr>
<td>TV</td>
<td>&lt;e,e&gt; \rightarrow t</td>
<td>transitive verbs (loves, sees)</td>
</tr>
</tbody>
</table>

- determine the semantic types of new expressions and vacuous words

2 Keynotes

- $\lambda$-calculus

- Semantic composition
3 Lambda calculus

• λ terms usually follow the schema below:

\[ \lambda \alpha : \phi. \gamma \]

a. \( \alpha \) is the argument variable (an arbitrary letter standing for the argument of the function we are defining)

b. \( \phi \) is the domain condition (the domain over which the function is defined).

c. \( \gamma \) is the value description (a specification of the value (or output) of the function we are defining.)

d. Read as:
   i. “the function which maps every \( \alpha \) such that \( \phi \) to \( \gamma \).”
   ii. “the function which maps every \( \alpha \) such that \( \phi \) to 1, if \( \gamma \), and to 0 otherwise.”

An example:

\[ \lambda x : x \in N. x + 1 \]

a. \( x \) is the argument variable.

b. \( x \in N \) is the domain condition (the set of natural numbers).

c. \( x + 1 \) is the value description.

d. read as “the function which maps every \( x \) such that \( x \) is in \( N \) to \( x + 1 \).”

\[ [\lambda x : x \in N. x + 1](2) = 3 \]

• Lexical entry of verbs:

(6)

a. \( [\text{snore}] = \lambda x : x \in D_e. x \text{ snores} \)

b. \( [\text{snore}] = \lambda x_e. x \text{ snores} \)

c. \( [\text{snore}](\text{John}) = [\lambda x : x \in D_e. x \text{ snores}](\text{John}) = 1 \) iff John smokes.

(7)

a. \( [\text{hit}] = \lambda y : y \in D_e. [\lambda x : x \in D_e. x \text{ hits } y] \)

b. \( [\text{hit}] = \lambda y_e. [\lambda x_e. x \text{ hits } y] \)

c. \( [\lambda y : y \in D_e. [\lambda x : x \in D_e. x \text{ hits } y](\text{Sue})] = \lambda y : y \in D_e. \text{Sue hit } y \)

d. \( [\lambda y : y \in D_e. [\lambda x : x \in D_e. x \text{ hits } y]](\text{Sue}) = [\lambda x : x \in D_e. x \text{ hit Sue}] \)

Discussion: To represent the lexical entry of ‘like’, we need to abstract the arguments \( x \) and \( y \) in \( [\text{like}(x, y)] \), consider, which argument should be abstracted first?

Exercise 1: Describe the following functions in English words:

(8)

a. \( \lambda x \in N. x > 3 \) and \( x < 7 \)

b. \( \lambda x : x \) is a person \( . \) \( x \)’s father
Exercise 2: Simplify the following functions as much as possible.

9) a. $[\lambda x \in D.[\lambda y \in D.[\lambda z \in D.z introduce x to y](Ann)](Sue)]$
   b. $[\lambda f \in D_{<e,t>}.[\lambda x \in D.e.f(x) = 1 and x is gray][[\lambda y \in D.e. y is a cat]]$
   c. $[\lambda x \in N.[\lambda y \in N.y > 3 and y < 7](x)]$

- Lexical entry of semantically vacuous words

10) a. $[of] = \lambda x \in D.e. x$
    b. $[be] = \lambda f \in D_{<e,t>} f$
    c. $[a] =$

- Lexical entry of non-verbal predicates

11) a. $[cat] =$
    b. $[red] =$
    c. $[fond] =$
    d. $[in] =$

Exercise 3:

1) Suppose “and” is of type $< t, < t, t >>$ and serves as a connective of two propositions. Specify this function using the $\lambda$-notation.

2) Suppose “and” is a connective of two VPs, what is the semantic type of “and”, and how to specify its lexicon?
Exercise 4: For each of the following λ-expressions, specify its semantic type.

(12) a. \[\lambda y \in D_e. \ y \text{ is a cat}\]
    b. \[\lambda y : y \in D_e. \ [\lambda x : x \in D_e. \ x \text{ hits } y]\]
    c. \[\lambda f \in D_{<e,t>}. [\lambda x \in D_e. f(x) = 1 \text{ and } x \text{ is gray}]
    d. \[\lambda f \in D_{<e,t>}. [\lambda g \in D_{<e,t>}. \text{there is some } x \in D_e \text{ such that } f(x) = 1 \text{ and } g(x) = 1]\]

Discussion: Translate the following sentences into predicate logic, consider, (i) how to specify the lexical entry of ‘student’ and ‘smart’? (ii) how to specify the lexical entry of ‘some’ and ‘every’ with λ-notation?

(13) a. Every student is smart.
    b. Some student is smart.
4 Tree diagram and semantic composition

- For each of the following sentences, use a tree diagram to illustrate its syntactic structure:

(14) a. John hit Bill.
    b. It is not the case John likes Sue.

- How to calculate denotations for these trees by means of composition rules and lexicon?

Composition rules

(15) **Terminal Nodes (TN)**  
If $\alpha$ is a terminal node, $[\alpha]$ is specified in the lexicon.

(16) **Non-Branching Nodes (NN)**  
If $\alpha$ is non-branching node, and $\beta$ is its daughter node, then $[\alpha] = [\beta]$.

(17) **Functional Application (FA)**  
If $\alpha$ is a branching node, $\{\beta, \gamma\}$ is the set of $\alpha$’s daughters, and $[\beta]$ is a function whose domain contains $[\gamma]$, $[\alpha] = [\beta](\gamma)$

- With the rules above, we are able to compose functions with their arguments, however, how to compose two items if they do not hold such a function-argument relation? E.g. restrictive modifier:

(18) Cambridge is [ a city [ in MA] ].

(19) **Predicate Modification (PM)**  
If $\alpha$ is a branching node, $\{\beta, \gamma\}$ is the set of $\alpha$’s daughters, and $[\beta]$ and $[\gamma]$ are both in $D_{<e,t>}$, then $[\alpha] = \lambda x \in D_e. \ [\beta](x) = [\gamma](x) = 1$
• An alternative to PM: to explore revised lexical entries for the words that may head modifiers and use functional application.

\[
\text{in} = \lambda y \in D_e. \ [\lambda f \in D_{<e,t>}. \ [\lambda x \in D_e. \ f(x) = 1 \text{ and } x \text{ is in } y]]
\]

**Discussion:** Now we have to options to analyze the following sentences: (i) analysis merely based on FA; (ii) analysis based on both FA and PM. Which method do you think is more plausible?

(21) a. Kitty is a gray cat.
    b. Jumbo is a small elephant.
    c. Ann is a fast speaker.

Suggested readings: Heim & Kratzer (1998) Chapter 3&4