Compositional Semantics

Roadmap

- Preliminary notions: Truth conditions, Extension
- (Extensional) Compositional Semantics
  - The principle of compositionality
  - Semantic types
  - Lambda calculus
  - Composition rules: TN, NN, FA, PM
- Generalized quantifiers

1 Truth conditions

- To know the meaning of a sentence is to know its **truth conditions**, namely, the knowledge of the conditions under which a sentence is true, and those under which it's false.

  (1) *Schema for truth conditions*
  The sentence "_____" is true if and only if _____.

  Example:

  (2) Mary lives in Cambridge is true if and only if Mary lives in Cambridge.

2 Extension

- The **extension** of an expression is dependent on the evaluation world. We add an *evaluation world parameter* $[\![\bullet]\!]^w$ to the notations of extensions:

  (3) General notation: $[\![X]\!]^w$ ('the extension of X in w')

  - $[\![\bullet]\!]$ is called **Interpretation function**; it maps syntactic expressions to their denotation/meaning.
  - The w in $[\![\bullet]\!]^w$ is called the evaluation world.

  Examples: (D_e denotes the set of entities)

  (4) a. $[\![\text{Mary lives in Cambridge}]\!]^w = 1$ iff Mary lives in Cambridge in w.
  b. $[\![\text{old}]\!]^w = \{x : x \text{ is old in } w\}$ (As a set)
  $[\![\text{old}]\!]^w = f : D_e \rightarrow \{0, 1\}$ such that
  for all $x \in D$, $f(x) = 1$ iff x is old in w. (As a characteristic function)

- The counterpart of *extension* is **intension**. The intension of X is a function which (i) takes a possible world as an argument, and (ii) returns the extension of X in that world. We will return to it later.

3 The principle of compositionality

- In any natural language, there are infinitely many sentences , and the brain is finite. So, for **syntax**, linguistic competence must involve some finitely describable means for specifying an infinite class of sentences.

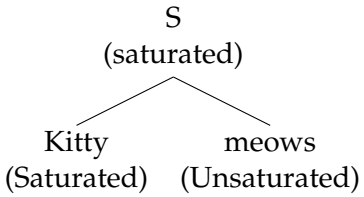
  A speaker of a language knows the meanings of those infinitely many sentences, and is able to understand a sentence he/she hears for the first time. So, for **semantics**, there must also be finite means for specifying the meanings of the infinite set of sentences of any natural language.

- In generative grammar, a central principle of formal semantics is that the relation between syntax and semantics is *compositional*.

  (5) **The principle of compositionality (Fregean Principle):**
  The meaning of a complex expression is determined by the meanings of its parts and the way they are syntactically combined.

The meaning of a sentence is the result of applying the unsaturated part of the sentence (a function) to the saturated part (an argument).

(6) Kitty meows.



– The meaning of the parts:

(7) a. $[\![\text{Kitty}]\!]^w = \text{Kitty}$
b. $[\![\text{meows}]\!]^w = \{x : x \text{ meows in } w\}$
c. $[\![\text{meows}]\!]^w = f : D_e \rightarrow \{1, 0\}$ such that
for every x: $f(x) = 1$ iff x meows in w.

– If we think of predicates as denoting sets of entities, then the composition of "Kitty" and "meows" proceeds via *set membership*:

(8) $[\![\text{Kitty meows}]\!]^w = 1$ iff $[\![\text{Kitty}]\!]^w \in [\![\text{meows}]\!]^w$
iff $\text{Kitty} \in \{x : x \text{ meows in } w\}$

– If we think of predicates as denoting functions (from sets of entities to truth values), then the composition of "Kitty" and "meows" proceeds via *functional application*:

(9) $[\![\text{Kitty meows}]\!]^w = [\![\text{meows}]\!]^w([\![\text{Kitty}]\!]^w)$
$= 1$ iff Kitty meows in w.

4 Type theory (for Semantics)

- The categories of syntax correspond in a one-to-one fashion to **semantic types**. The basic types correspond to the objects that Frege takes to be saturated.
  - e for individuals, in D_e
  - t for truth values, in $\{1, 0\}$

  From these basic types, we can recursively define complex types:
  - $\langle e, t\rangle$ for intransitive verbs, predicative adjectives, and common nouns
  - $\langle e, \langle e, t\rangle\rangle$ for transitive verbs

(10) **Types**
  a. **Basic types**: e (individuals/entities) and t (truth values).
  b. **Functional types**: If α and β are types, then $\langle \alpha, \beta\rangle$ is a type. A function of type $\langle \alpha, \beta\rangle$ is one whose arguments/inputs are of type α and whose values/outputs are of type β.

(11) **Domains**
  a. $D_t = \{1, 0\}$
  b. $D_e = \{x : x \text{ is an entity}\}$
  c. $D_{\langle \alpha,\beta\rangle} = \{f \mid f : D_\alpha \rightarrow D_\beta\}$ (functions from things of type α to things of type β)

- Syntactic categories and their semantic types (an inclusive list)

Syntactic category	**English expressions**	**Semantic type** (extensionalized)
Sentence		t
Proper name	*John*	e
e-type/referential NP	*the king*	e
Common noun	*cat*	$\langle e, t\rangle$
IV, VP	*run*	$\langle e, t\rangle$
TV	*love*, *buy*	$\langle e, et\rangle$
Predicative ADJ	*happy*, *gray*	$\langle e, t\rangle$
Predicate modifier	*skillful*, *quickly*	$\langle et, et\rangle$
Sentential modifier	*perhaps*	$\langle t, t\rangle$
Determiner	*some*, *every*, *no*	$\langle et, \langle et, t\rangle\rangle$
	the	$\langle et, e\rangle$

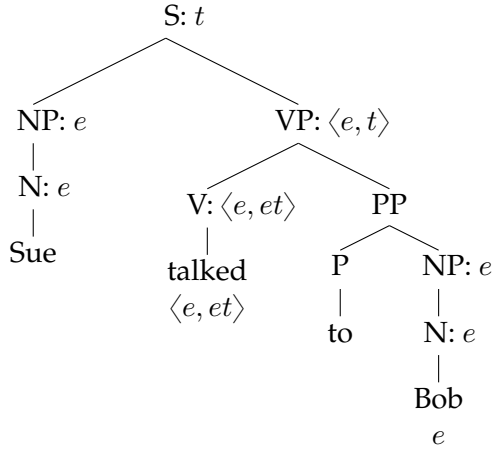
Exercise: Identify the semantic type of *beautiful* in the following sentences:

(12) a. Jenny is a *beautiful* girl.
  b. Jenny is a *beautiful* dancer.

Exercise: Classify the following words based on their semantic types:
not, *if...then*, *student*, *John*, *Boston*, *a man*, *buy*, *fast*, *carefully*, *necessarily*

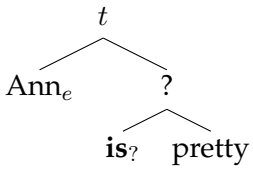
- In most cases, with type assignments to expressions of natural language, we can determine the semantic types of new expressions/morphemes.

(13) Susan talked <u>to</u> Bob.

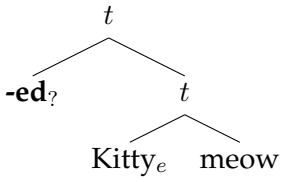


Exercise: Identify the semantic types of the underlined words/morphemes.

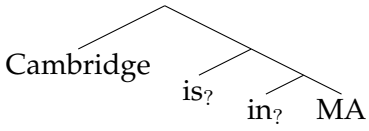
(14) Ann <u>is</u> pretty.



(15) Kitty meow<u>ed</u>.



(16) Cambridge <u>is</u> <u>in</u> MA.



- But, be careful!

(i) some expressions can be type-flexible;

(ii) there can be covert elements in the LF;

(iii) there can be type-shifting operations;

(iv) two sister nodes might not hold a function-argument relation; ...