Semantic Types and Function Application

Ling324

Reading: *Meaning and Grammar*, pg. 87-98

Semantic Types we have specified so far for the fragment of English F1

Syntactic Category	Semantic Type
S	Truth values (0 or 1)
N	Individuals
V$_i$, VP	Sets of individuals
V$_t$	Sets of ordered pairs of individuals
Conj	Function from pairs of truth values to truth values
Neg	Function from truth values to truth values

Specifying Semantic Rules in terms of Function Application

- A function takes an input argument from some specified domain and yields an output value.

  Applying a function f to an argument x yields the value for that argument, which can be written as $f(x)$.

  The mode of combining a function and its argument is called FUNCTION APPLICATION.

- The way we have defined the semantics of Neg makes use of function application.

  $[\![\text{Neg}]\!]^V = \begin{bmatrix} 1 \rightarrow 0 \\ 0 \rightarrow 1 \end{bmatrix}$ = the function f from truth values to truth values such that: $f(1) = 0$ and $f(0) = 1$

- In fact, function application could be used to interpret any syntactic structure with two branches: one branch is interpreted as a function, and the other branch is interpreted as a possible argument of the function.

  $$[\![\ [_A\ B\ C]\ ]\!]^V = [\![B]\!]^V([\![C]\!]^V)$$

Intransitive Verb

- $[\![\text{is cute}]\!]^V = \{x : x \text{ is cute in } V\}$

  For example, let Universe = $\{$Fiona, Patsy, Jenny, John$\}$
  $[\![\text{is cute}]\!]^V$ = $\{$Fiona, Jenny$\}$

- $[\![\text{is cute}]\!]^V$ = the function f from individuals to truth values such that:
  $f(x)$ = 1 if $x \in \{x : x \text{ is cute in } V\}$, and $f(x)$ = 0 otherwise.

  $$[\![\text{is cute}]\!]^V = \begin{bmatrix} \text{Fiona} \rightarrow 1 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 1 \\ \text{John} \rightarrow 0 \end{bmatrix}$$ (= the characteristic function of $\{x : x \text{ is cute in } V\}$)

- Characteristic function

  Any function that assigns one of two distinct values (0 or 1) to the members of a domain is called CHARACTERISTIC FUNCTION.

  Each subset of the domain defines such a function uniquely, and any such function corresponds to a unique subset of the domain.

  This means that we can use sets and characteristic function of that set interchangeably when defining the semantic value of intransitive verbs.

Intransitive Verb (cont.)

- Semantic types

  e (entity): the type of individuals.

  t (truth value): the type of truth values.

  $< e, t >$: the type of functions from individuals into truth values.

- Intransitive verb combines with the subject, by function application, and returns a truth value.

  $[\![\text{is cute}]\!]^V([\![\text{John}]\!]^V)$ = 1 or 0 (depending on the situation V)

- QUESTION: Provide the semantic value for *is hungry* and *is boring* in terms of set notation, and functional notation.

Transitive Verb

- $[\![\text{likes}]\!]^V = \{< x, y >: x \text{ likes } y \text{ in } V\}$

  For example, let Universe = $\{$Fiona, Patsy, Jenny$\}$
  $[\![\text{likes}]\!]^V = \{<\text{Fiona, Patsy}>, <\text{Patsy, Jenny}>, <\text{Jenny, Jenny}>\}$

- The characteristic function of $[\![\text{likes}]\!]^V$

$$\begin{bmatrix} <\text{Fiona, Fiona}> \rightarrow 0 \\ <\text{Fiona, Patsy}> \rightarrow 1 \\ <\text{Fiona, Jenny}> \rightarrow 0 \\ <\text{Patsy, Fiona}> \rightarrow 0 \\ <\text{Patsy, Patsy}> \rightarrow 0 \\ <\text{Patsy, Jenny}> \rightarrow 1 \\ <\text{Jenny, Fiona}> \rightarrow 0 \\ <\text{Jenny, Patsy}> \rightarrow 0 \\ <\text{Jenny, Jenny}> \rightarrow 1 \end{bmatrix}$$

Transitive Verb (cont.)

- Schönfinkelization: Turning n-ary functions into multiple embedded unary functions.

Left-to-right

$$\begin{bmatrix} \text{Fiona} \rightarrow \begin{bmatrix} \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 1 \\ \text{Jenny} \rightarrow 0 \end{bmatrix} \\ \text{Patsy} \rightarrow \begin{bmatrix} \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 1 \end{bmatrix} \\ \text{Jenny} \rightarrow \begin{bmatrix} \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 1 \end{bmatrix} \end{bmatrix}$$

Right-to-left

$$\begin{bmatrix} \text{Fiona} \rightarrow \begin{bmatrix} \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 0 \end{bmatrix} \\ \text{Patsy} \rightarrow \begin{bmatrix} \text{Fiona} \rightarrow 1 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 0 \end{bmatrix} \\ \text{Jenny} \rightarrow \begin{bmatrix} \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 1 \\ \text{Jenny} \rightarrow 1 \end{bmatrix} \end{bmatrix}$$

- Which Schönfinkelization is consistent with the principle of compositional semantics? Left-to-right or Right-to-left?

Transitive Verb (cont.)

- $[\![\text{likes}]\!]^V$ = the function f from individuals to characteristic functions such that: $f(y) = g_y$, the characteristic function of $\{x : x \text{ likes } y \text{ in } V\}$.

- Type of functions from individuals to characteristic functions

  $< e, < e, t >>$

- Transitive verb combines with a direct object, by function application, and returns a characteristic function of a set.

  $[\![\text{likes}]\!]^V([\![\text{Vivian}]\!]^V)$ = the function f from individuals to truth values such that: $f(x) = 1$ if $x \in \{x : x \text{ likes Vivian in } V\}$, and $f(x) = 0$ otherwise.

Logical Connectives: *and*

- Binary function

$$\begin{bmatrix} <1,1>\rightarrow 1 \\ <1,0>\rightarrow 0 \\ <0,1>\rightarrow 0 \\ <0,0>\rightarrow 0 \end{bmatrix}$$

- Schönfinkelization

  Assume the following two syntactic rules:

  (1) a. S $\longrightarrow$ S conjP
      b. conjP $\longrightarrow$ conj S

- Unary function

$$\begin{bmatrix} 1 \rightarrow \begin{bmatrix} 1 \rightarrow 1 \\ 0 \rightarrow 0 \end{bmatrix} \\ 0 \rightarrow \begin{bmatrix} 1 \rightarrow 0 \\ 0 \rightarrow 0 \end{bmatrix} \end{bmatrix}$$

- Type of functions from truth values to functions from truth values to truth values

  $< t, < t, t >>$

Calculating Truth Conditions using Functional Approach

- Semantic Rules

(2) a. Pass-up:
If Δ is a nonbranching node that dominates a,
then $[\![\Delta]\!]^V = [\![a]\!]^V$

b. Function Application
If Δ is a branching node with daughters a and b,
and $[\![a]\!]^V$ is a function whose domain contains $[\![b]\!]^V$,
then $[\![\Delta]\!]^V = [\![a]\!]^V([\![b]\!]^V)$.

- EXERCISE: For the following examples, calculate their truth conditions compositionally using the semantic rules above.

(3) a. Bob is hungry.

b. Kitty likes Vivian.

Specifying Semantic Types in terms of Functional Types

Syntactic Category	Semantic Type
S	t
N	e
V_i, VP	$< e, t >$
V_t	$< e, < e, t >>$
Conj	$< t, < t, t >>$
Neg	$< t, t >$