

# 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.

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

- 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.

$$\llbracket \begin{array}{c} A \\ B \quad C \end{array} \rrbracket^V = \llbracket B \rrbracket^V (\llbracket C \rrbracket^V)$$

## Intransitive Verb

- $\llbracket \text{is cute} \rrbracket^V = \{x : x \text{ is cute in } V\}$

For example, let Universe = {Fiona, Patsy, Jenny, John}

$$\llbracket \text{is cute} \rrbracket^V = \{\text{Fiona, Jenny}\}$$

- $\llbracket \text{is cute} \rrbracket^V$  = 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.

$$\llbracket \text{is cute} \rrbracket^V = \left[ \begin{array}{l} \text{Fiona} \rightarrow 1 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 1 \\ \text{John} \rightarrow 0 \end{array} \right] \quad (= \text{the characteristic function of } \{x : x \text{ is cute in } V\})$$

- 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.

$\langle e, t \rangle$ : the type of functions from individuals into truth values.

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

$\llbracket \text{is cute} \rrbracket^V (\llbracket \text{John} \rrbracket^V) = 1 \text{ or } 0$  (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

- $[[\text{likes}]]^V = \{ \langle x, y \rangle : x \text{ likes } y \text{ in } V \}$

For example, let Universe = {Fiona, Patsy, Jenny}

$$[[\text{likes}]]^V = \{ \langle \text{Fiona}, \text{Patsy} \rangle, \langle \text{Patsy}, \text{Jenny} \rangle, \langle \text{Jenny}, \text{Jenny} \rangle \}$$

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

$$\left[ \begin{array}{l} \langle \text{Fiona}, \text{Fiona} \rangle \rightarrow 0 \\ \langle \text{Fiona}, \text{Patsy} \rangle \rightarrow 1 \\ \langle \text{Fiona}, \text{Jenny} \rangle \rightarrow 0 \\ \langle \text{Patsy}, \text{Fiona} \rangle \rightarrow 0 \\ \langle \text{Patsy}, \text{Patsy} \rangle \rightarrow 0 \\ \langle \text{Patsy}, \text{Jenny} \rangle \rightarrow 1 \\ \langle \text{Jenny}, \text{Fiona} \rangle \rightarrow 0 \\ \langle \text{Jenny}, \text{Patsy} \rangle \rightarrow 0 \\ \langle \text{Jenny}, \text{Jenny} \rangle \rightarrow 1 \end{array} \right]$$

## Transitive Verb (cont.)

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

Left-to-right

$$\left[ \begin{array}{l} \text{Fiona} \rightarrow \\ \text{Patsy} \rightarrow \\ \text{Jenny} \rightarrow \end{array} \left[ \begin{array}{l} \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 1 \\ \text{Jenny} \rightarrow 0 \\ \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 1 \\ \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 1 \end{array} \right] \right]$$

Right-to-left

$$\left[ \begin{array}{l} \text{Fiona} \rightarrow \\ \text{Patsy} \rightarrow \\ \text{Jenny} \rightarrow \end{array} \left[ \begin{array}{l} \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 0 \\ \text{Fiona} \rightarrow 1 \\ \text{Patsy} \rightarrow 0 \\ \text{Jenny} \rightarrow 0 \\ \text{Fiona} \rightarrow 0 \\ \text{Patsy} \rightarrow 1 \\ \text{Jenny} \rightarrow 1 \end{array} \right] \right]$$

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

## Transitive Verb (cont.)

- $\llbracket \text{likes} \rrbracket^V$  = 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

$\langle e, \langle e, t \rangle \rangle$

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

$\llbracket \text{likes} \rrbracket^V (\llbracket \text{Vivian} \rrbracket^V)$  = 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

$$\left[ \begin{array}{l} \langle 1, 1 \rangle \rightarrow 1 \\ \langle 1, 0 \rangle \rightarrow 0 \\ \langle 0, 1 \rangle \rightarrow 0 \\ \langle 0, 0 \rangle \rightarrow 0 \end{array} \right]$$

- Schönfinkelization

Assume the following two syntactic rules:

- (1)    a.      $S \rightarrow S \text{ conj} P$   
      b.      $\text{conj} P \rightarrow \text{conj } S$

- Unary function

$$\left[ \begin{array}{l} 1 \rightarrow \left[ \begin{array}{l} 1 \rightarrow 1 \\ 0 \rightarrow 0 \end{array} \right] \\ 0 \rightarrow \left[ \begin{array}{l} 1 \rightarrow 0 \\ 0 \rightarrow 0 \end{array} \right] \end{array} \right]$$

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

$$\langle t, \langle t, t \rangle \rangle$$

# Calculating Truth Conditions using Functional Approach

- Semantic Rules

- (2) a. Pass-up:

- If  $\Delta$  is a nonbranching node that dominates  $a$ ,  
then  $\llbracket \Delta \rrbracket^V = \llbracket a \rrbracket^V$

- b. Function Application

- If  $\Delta$  is a branching node with daughters  $a$  and  $b$ ,  
and  $\llbracket a \rrbracket^V$  is a function whose domain contains  $\llbracket b \rrbracket^V$ ,  
then  $\llbracket \Delta \rrbracket^V = \llbracket a \rrbracket^V(\llbracket b \rrbracket^V)$ .

- 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	$\langle e, t \rangle$
$V_t$	$\langle e, \langle e, t \rangle \rangle$
Conj	$\langle t, \langle t, t \rangle \rangle$
Neg	$\langle t, t \rangle$