

1 Warm-up

1. Give the semantic type of each of the following functions expressed in lambda notation. Some of these might not correspond to an actual linguistic expression, but are functions nonetheless. (d = Albus Dumbledore) **Solution given in class**

- (a) $\lambda z[BUBOTUBER(z)]$
- (b) $\lambda y[\lambda x[HEX(x, y)]]$
- (c) $\exists x[QUAFFLE(x) \ \& \ RED(x)]$
- (d) $\lambda x[IMPRESS(x, d)]$
- (e) d
- (f) $\lambda x[d]$

2. Give the denotation of the underlined word in lambda notation. State the semantic type of it as well. **Solution given in class**

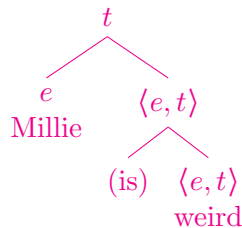
- (a) Fred coughed
- (b) Rupert is a dog
- (c) Peter saw Katrina
- (d) Bob is in Tokyo

3. Give a full lambda computation of the following sentences, including a tree annotated with types, the lexical entries (i.e., give me the denotation of each word, using lambda notation where needed), and a step-by-step computation. You can ignore tense, and you can treat **a**, *is*, and *of* as meaningless.

- (a) Millie is weird

$\llbracket \text{Millie} \rrbracket = m$

$\llbracket \text{weird} \rrbracket = \lambda x[\text{WEIRD}(x)]$



$\llbracket \text{Millie (is) weird} \rrbracket$

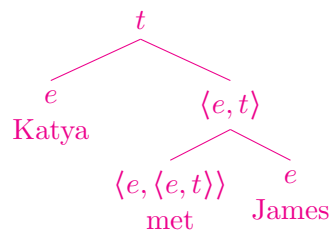
i. $= \llbracket \text{weird} \rrbracket (\llbracket \text{Millie} \rrbracket)$

ii. $= \llbracket \text{weird} \rrbracket (m)$

iii. $= \lambda x[\text{WEIRD}(x)](m)$

iv. $= \text{T iff WEIRD}(m)$

(b) Katya met James

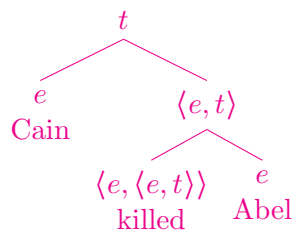
 $\llbracket \text{Katya} \rrbracket = k$ $\llbracket \text{James} \rrbracket = j$ $\llbracket \text{meet} \rrbracket = \lambda x[\lambda y[\text{MEET}(y, x)]]$  $\llbracket \text{Katya met James} \rrbracket$ i. $\llbracket \text{met James} \rrbracket$ A. $= \llbracket \text{meet James} \rrbracket$ B. $= \llbracket \text{meet} \rrbracket (\llbracket \text{James} \rrbracket)$ C. $= \llbracket \text{meet} \rrbracket (j)$ D. $= \lambda x[\lambda y[\text{MEET}(y, x)]](j)$ E. $= \lambda y[\text{MEET}(y, j)]$ ii. $\llbracket \text{Katya met James} \rrbracket$ A. $= \llbracket \text{Katya meet James} \rrbracket$ B. $= \llbracket \text{meet James} \rrbracket (\llbracket \text{Katya} \rrbracket)$ C. $= \llbracket \text{meet James} \rrbracket (k)$ D. $= \lambda y[\text{MEET}(y, j)](k)$ E. $= \text{T iff MEET}(k, j)$

(c) Cain killed Abel

$\llbracket \text{Cain} \rrbracket = c$

$\llbracket \text{Abel} \rrbracket = a$

$\llbracket \text{kill} \rrbracket = \lambda x[\lambda y[\text{KILL}(y, x)]]$



$\llbracket \text{Cain killed Abel} \rrbracket$

i. $\llbracket \text{killed Abel} \rrbracket$

A. $= \llbracket \text{kill Abel} \rrbracket$

B. $= \llbracket \text{kill} \rrbracket (\llbracket \text{Abel} \rrbracket)$

C. $= \llbracket \text{kill} \rrbracket (a)$

D. $= \lambda x[\lambda y[\text{KILL}(y, x)]](a)$

E. $= \lambda y[\text{KILL}(y, a)]$

ii. $\llbracket \text{Cain killed Abel} \rrbracket$

A. $= \llbracket \text{Cain killed Abel} \rrbracket$

B. $= \llbracket \text{killed Abel} \rrbracket (\llbracket \text{Cain} \rrbracket)$

C. $= \llbracket \text{killed Abel} \rrbracket (c)$

D. $= \lambda y[\text{KILL}(y, a)](c)$

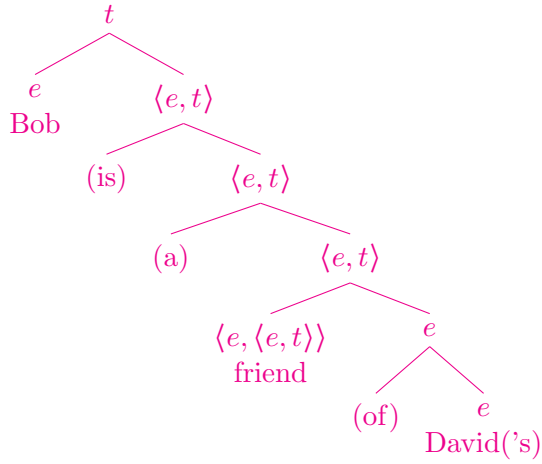
E. $= \text{T iff KILL}(c, a)$

(d) Bob is a friend of David('s)

$\llbracket \text{Bob} \rrbracket = b$

$\llbracket \text{David} \rrbracket = d$

$\llbracket \text{friend} \rrbracket = \lambda x[\lambda y[\text{FRIEND}(y, x)]]$



$\llbracket \text{Bob (is) (a) friend (of) David('s)} \rrbracket$

i. $\llbracket (\text{is}) (\text{a}) \text{friend (of) David('s)} \rrbracket$

(A. = $\llbracket \text{friend David} \rrbracket$)

B. = $\llbracket \text{friend} \rrbracket (\llbracket \text{David} \rrbracket)$

C. = $\llbracket \text{friend} \rrbracket (d)$

D. = $\lambda x[\lambda y[\text{FRIEND}(y, x)]](d)$

E. = $\lambda y[\text{FRIEND}(y, d)]$

[you can show this step if you want]

ii. $\llbracket \text{Bob (is) (a) friend (of) David('s)} \rrbracket$

(A. $\llbracket \text{Bob friend David} \rrbracket$)

B. = $\llbracket (\text{is}) (\text{a}) \text{friend (of) David('s)} \rrbracket (\llbracket \text{Bob} \rrbracket)$

C. = $\llbracket (\text{is}) (\text{a}) \text{friend (of) David('s)} \rrbracket (b)$

D. = $\lambda y[\text{FRIEND}(y, d)](b)$

E. = T iff $\text{FRIEND}(b, d)$