

# Modification and translation

## 1 Review of rules

(1) **Terminal Nodes (TN):**

If  $\alpha$  is a terminal node,  $\llbracket \alpha \rrbracket$  is specified in the lexicon.

(2) **Non-branching Nodes (NN):**

If  $\alpha$  is a non-branching node, and  $\beta$  is its daughter node, then  $\llbracket \alpha \rrbracket = \llbracket \beta \rrbracket$ .

(3) **Functional Application (FA):**

If  $\alpha$  is a branching node,  $\{\beta, \gamma\}$  is the set of  $\alpha$ 's daughters, and  $\llbracket \beta \rrbracket$  is a function whose domain contains  $\llbracket \gamma \rrbracket$ , then  $\llbracket \alpha \rrbracket = \llbracket \beta \rrbracket(\llbracket \gamma \rrbracket)$ .

## 2 How to study the meaning of a part

Using the Principle of Compositionality, we can figure out the meaning of individual parts of sentences.

- (4) Kara **and** Tama sleep.
- (5) John likes **himself**.
- (6) Sarah swims **again**.

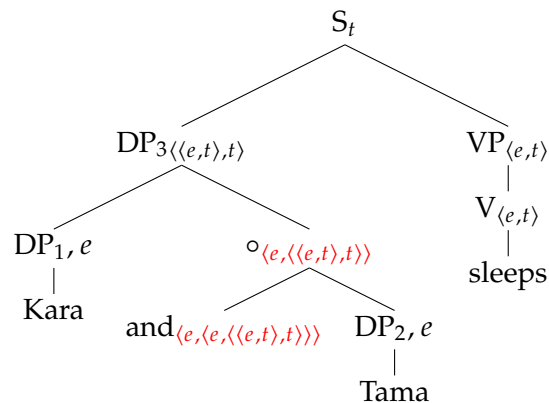
### Step by step:

1. What does the whole sentence mean? Paraphrase without using the target part (in bold).
2. What is the structure of the sentence? Draw a tree.
3. Fill in semantic types. Use the Triangle Method if necessary.
4. Using your paraphrase from Step 1, work backwards to figure out the meaning of the target part (in bold).
  - Make sure the meaning you write for the target part is general: it should not include meanings which are contributed from other material in the sentence.
  - Remember that each  $\lambda$  should correspond to a variable in the return value. When you add a  $\lambda$  variable, make sure it's used.
5. Check that your final meaning matches the predicted type. Recompute the structure bottom-up to make sure it works. Make sure the meaning you proposed also works in other, similar examples.

**Example:**

(4) Kara and Tama sleep.

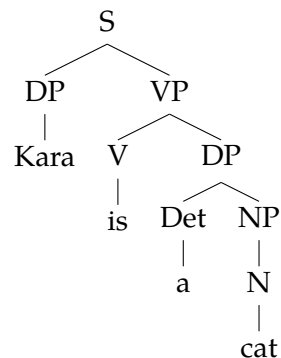
First, to figure out the types. The important thing to note is that there is no plural “Kara+Tama” in  $D_e$ . This teaches us that the type of the DP “Kara and Tama” cannot be type  $e$ . The only other option (using the Triangle Method, using Functional Application) is type  $\langle\langle e, t \rangle, t\rangle$ . Our goal is to figure out a way to get (3) to mean the same thing as “Kara sleeps and Tama sleeps.”



- $\llbracket \text{VP} \rrbracket_{\text{NN}} = \llbracket \text{sleep} \rrbracket_{\text{TN}} = \lambda x_e . \text{Sleep}(x)$
- $\llbracket \text{DP}_1 \rrbracket_{\text{TN}} = \text{Kara}$
- $\llbracket \text{DP}_2 \rrbracket_{\text{TN}} = \text{Tama}$
- Definition of *and*:  $\llbracket \text{and} \rrbracket_{\text{TN}} = \lambda x_e . \lambda y_e . \lambda P_{\langle e, t \rangle} . P(x) \wedge P(y)$
- $\llbracket \circ \rrbracket_{\text{FA}} = \llbracket \text{and} \rrbracket (\llbracket \text{DP}_2 \rrbracket)$   
 $= [\lambda x_e . \lambda y_e . \lambda P_{\langle e, t \rangle} . P(x) \wedge P(y)] (\text{Tama})$   
 $= \lambda y_e . \lambda P_{\langle e, t \rangle} . P(\text{Tama}) \wedge P(y)$
- $\llbracket \text{DP}_3 \rrbracket_{\text{FA}} = \llbracket \circ \rrbracket (\llbracket \text{DP}_1 \rrbracket)$   
 $= [\lambda y_e . \lambda P_{\langle e, t \rangle} . P(\text{Tama}) \wedge P(y)] (\text{Kara})$   
 $= \lambda P_{\langle e, t \rangle} . P(\text{Tama}) \wedge P(\text{Kara})$
- $\llbracket \text{S} \rrbracket_{\text{FA}} = \llbracket \text{DP}_3 \rrbracket (\llbracket \text{VP} \rrbracket)$   
 $= [\lambda P_{\langle e, t \rangle} . P(\text{Tama}) \wedge P(\text{Kara})] (\lambda x_e . \text{Sleep}(x))$   
 $= 1 \text{ iff } (\lambda x_e . \text{Sleep}(x))(\text{Tama}) \wedge (\lambda x_e . \text{Sleep}(x))(\text{Kara})$   
 $= 1 \text{ iff } \text{Sleep}(\text{Tama}) \wedge \text{Sleep}(\text{Kara})$

### 3 Non-verbal predicates

(7) Kara is a cat.



Compositionality allows us to (a) use what we know and (b) work backwards.

(8) Kara sleeps and is a cat.

The semantics for conjunction developed in PS3 (hopefully) is only defined for conjunctions of equal semantic type.

- (9)
- a. Austin is a city and Austin is in Texas.
  - b. Austin is a city and is in Texas.
  - c. Austin is a city and in Texas.
  - d. \* Austin is a city and Texas.

### 4 Modification

- (10)
- a. Kara is a black cat.
  - b. Kara is black and Kara is a cat.
- (11)
- a. Austin is a city in Texas.
  - b. Austin is a city and Austin is in Texas.

Each pair of sentences in (10a,b) and (11a,b) is truth-conditionally equivalent. We call such modifiers *intersective*.

Option 1: Intuitively, *black* modifies *cat*. Write a semantics so that  $\llbracket \text{black} \rrbracket$  modifies  $\llbracket \text{cat} \rrbracket$  through Functional Application.

$$(12) \quad \llbracket \text{black} \rrbracket = \lambda P_{\langle e, t \rangle} . \lambda x . \text{Black}(x) \wedge P(x)$$

The disadvantage of this approach is that attributive adjectives (modifiers) and predicate adjectives have different semantics, although taking a predicate adjective  $\langle e, t \rangle$  and converting it to its attributive form  $\langle \langle e, t \rangle, \langle e, t \rangle \rangle$  is easy: *IFS* page 195 describes this as MOD.

Option 2: Introduce a new composition rule.

(13) **Predicate Modification (PM):** (also in *IFS*: 196)  
If  $\alpha$  is a branching node,  $\{\beta, \gamma\}$  is the set of  $\alpha$ 's daughters, and  $\llbracket \beta \rrbracket$  and  $\llbracket \gamma \rrbracket$  are both in  $D_{\langle e, t \rangle}$ , then  $\llbracket \alpha \rrbracket = \lambda x_e . \llbracket \beta \rrbracket (x) \wedge \llbracket \gamma \rrbracket (x)$ .

Now we can simply use the regular  $\langle e, t \rangle$  denotations for *black* and *in Texas*.

## 5 Non-intersective modifiers

What about the following modifiers?

- (14) a. This is a fake diamond.  
b. This is fake and is a diamond.
- (15) a. John is a short basketball-player.  
b. This is short and is a basketball-player.
- (16) a. Obama is a former president.  
b. \*Obama is former and is a president.