LING 130: Guide to Problem Set 3

James Pustejovsky

March 10, 2010

1 Keeping Track of Substitution Values

Recall that the method of *Quanitifier Substitution* has two components to it.

(1) a. Quantifier Substitution
b. Substitution Application

The first rule applies when a functional expression, α, is requesting an argument of a certain type, and the potential argument expression is not of that type. For example, if α is looking for an entity type, e, but the other expression, β is a generalized quantifier, $(e \rightarrow t) \rightarrow t$, then perform a substitution of β with a constant$C$, that satisfies the type requested by α; $[C/\beta]$. Generally, this rule is stated as follows:

(2) Quantifier Substitution:
For every expression, γ, in a sentence, we associate a body, α, and the set of quantifier substitutions, Σ, where Σ = $\{[C_1/Q_1]_{\sigma_1}, [C_2/Q_2]_{\sigma_2}, \ldots, [C_n/Q_n]_{\sigma_n}\}$
$\gamma = \alpha\{\Sigma\}$

The second part of the method is a rule called *Substitution Application*. This applies to each substitution, σ_i in Σ, and it performs the following operation:

(3) $\alpha\{\sigma_u\} \Longrightarrow \sigma_u(\lambda u \alpha[u])$

Let's see how this method works when the sentence has more than one QNP in it, such as those in (4)-(5) below.

(4) a. A judge sentenced every prisoner.
b. $\forall x[prisoner(x) \rightarrow \exists y[judge(y) \wedge sentence(y, x)]]$
c. $\exists y[judge(y) \wedge \forall x[prisoner(x) \rightarrow sentence(y, x)]]$

(5) a. Every dog ate a bone.
b. $\forall x[dog(x) \rightarrow \exists y[bone(y) \wedge eat(x, y)]]$
c. $\exists y[bone(y) \wedge \forall x[dog(x) \rightarrow eat(x, y)]]$

Here's how to think about this problem. How many quantifier NPs are there? For each one, you will need a substitution. Since they are independent of one another (e.g., there's no embedded quantifiers), we can picture how each one gets substituted by the QS method. Consider the two quantifiers in (4).

(6) a. ⟨"every prisoner", $(e \to t) \to t$, $\lambda P \forall x[prisoner(x) \to P(x)]$⟩
b. QS: ⟨ "every prisoner", e, C_1⟩, $[C_1/\lambda P \forall x[prisoner(x) \to P(x)]]_{\sigma_1}$

(7) a. ⟨"a judge", $(e \to t) \to t$, $\lambda P \exists x[judge(x) \land P(x)]$⟩
b. QS: ⟨ "a judge", e, C_2⟩, $[C_2/\lambda P \exists x[judge(x) \land P(x)]]_{\sigma_2}$

So now, let's go through each interpretation in (4), starting with the wide-scope on *a judge*. That is, there is one judge that sentenced all the prisoners.

(8) STEP-BY-STEP:
a. A judge sentenced every prisoner.
b. sentence: $\lambda x \lambda y[sentence(y, x)]$
c. Quantifier Substitution (QS): $C_1 : e, [C_1/\lambda P \forall x[prisoner(x) \to P(x)]]_{\sigma_1}$
d. Function Application: $\lambda x \lambda y[sentence(y, x)] : e \to (e \to t), C_1 : e \Longrightarrow$
$\lambda y[sentence(y, C_1)]\{\sigma_1\} : e \to t$
e. Quantifier Substitution (QS):
$C_2 : e, [C_2/\lambda P \exists x[judge(x) \land P(x)]]_{\sigma_2}$
f. Function Application: $\lambda y[sentence(y, C_1)]\{\sigma_1\} : e \to t, C_2 : e \Longrightarrow$
$[sentence(C_2, C_1)]\{\sigma_1, \sigma_2\} : t$
g. Substitution Application on σ_1:
$\lambda P \forall x[prisoner(x) \to P(x)]\{\sigma_2\}(\lambda z[sentence(C_2, z)])$
h. Function Application:
$\forall x[prisoner(x) \to sentence(C_2, x)]\{\sigma_2\}$
i. Substitution Application on σ_2:
$\lambda P \exists y[judge(y) \land P(y)](\lambda w \forall x[prisoner(x) \to sentence(w, x)])$
j. Function Application:
$\exists y[judge(y) \land \forall x[prisoner(x) \to sentence(y, x)]]$
o. ☺

The other reading starts with the above derivation at (9f), and applies SA on σ_2 first, then moves on to SA for σ_1.

(9) STEP-BY-STEP:
a. A judge sentenced every prisoner.
...
f. Function Application: $\lambda y[sentence(y, C_1)]\{\sigma_1\} : e \to t, C_2 : e \Longrightarrow$
$[sentence(C_2, C_1)]\{\sigma_1, \sigma_2\} : t$
g. Substitution Application on σ_2:
$\lambda P \exists x[judge(x) \land P(x)]\{\sigma_1\}(\lambda z[sentence(z, C_1)])$
h. Function Application:
$\exists x[judge(x) \land sentence(x, C_1)]\{\sigma_1\}$
i. Substitution Application on σ_1:
$\lambda P \forall y[prisoner(y) \to P(y)](\lambda w \exists x[judge(x) \land sentence(x, w)])$
j. Function Application:
$\forall y[prisoner(y) \to \exists x[judge(x) \land sentence(x, y)]]$
o. ☺