Assignment 3 Tutorial 3

Jinyue Feng

Overview

- TRALE Basics
- Grammatical Gender Agreement
- Subcategorization
- Passive Voice (Gap Constuct)
- Semantic Head
- Goals and Macros
- Beta Reduction
- Quantifier Storage

Don't forget to check out <u>the tutorial online</u>!

Macros and Goals - Practical Aspects in Q2

- Use the macros to translate the formal logic into TRALE
 - @lambda(X, F): $\lambda X.F$
 - @forall(X, F, G): $\forall X.F \Rightarrow G$
 - \circ @exists(X, F, G): $\exists X.F \land G$
 - @apply(F, [X, Y, Z]): F(X,Y,Z)
- Use goals to apply the proper process at given parsing step.
 - Beta Reduction
 - goal > beta_normalize(@apply(F, [G]), B).
 - (B is β normalized F (G))
 - Quantifier Storage
 - goal > qaction(Logic, Qstore, NewLogic, NewQstore).
 - Quantifier Retrieval
 - goal > retrieve(Qstore, Logic, NewQstore, NewLogic).

Macros and Goals - Practical Aspects in Q2

- Encoding course(x): use @apply
 - In the last tutorial, we saw goal-style definition

```
course(x) if true.
```

Given the definition of our helper function:

beta_normalize((app,Apply),Apply) if true.

By using statements like <code>@lambda(@apply(Course, [x])</code>, we can fulfill the goal in the exact same way without explicitly defining separate goals for each type of semantic.

Notice that Apply is a variable, not an atom.

Beta Reduction

Every student takes a course.





Every student takes a course.

How do we go from the surface reading:

 $\forall x.(student(x) \Rightarrow \exists y.(course(y) \land take(x, y)))$

To:

 $\exists y.(course(y) \land \forall x.(student(x) \Rightarrow take(x, y)))$

Basic Idea of Quantifier Storage

- Similar to the idea of subcategorization, we use a list (qstore) to save items (in this case, logic) for future processing.
 - We are switching the order of processing to reach the second reading.
- A placeholder function takes the place of the stored logic and participates in the beta normalization steps.
- When it is time to process the stored logic, we apply the stored function into the placeholder and beta normalize the final expression.
 - Note that our retrieve goal already contains the beta_normalization goal as a condition.

Quantifier storage



Storage and Retrieval



What happens at goal> qaction?

NP: A course

	Logic	Qstore
before	$\lambda G. \exists y. (course(y) \land G(y))$	<>
after	$\lambda F.F(z)$	$\langle z; \lambda G. \exists y. (language(y) \land G(y)) \rangle$

What happens at goal> retrieve?

S: Every student takes a course

	Logic	Qstore
Before	$\forall x.student(x) \Rightarrow take(x, z)$ Notice this is already beta-normalized	<z;λg.∃y.(language(y)∧g(y))></z;λg.∃y.(language(y)∧g(y))>
Step 1	$\lambda z. \forall x. student(x) \Rightarrow take(x, z)$	$\langle z; \lambda G. \exists y. (language(y) \land G(y)) \rangle$
Step 2	$(\lambda G.\exists y.(course(y) \land G(y)))$ $(\lambda z.\forall x.student(x) \Rightarrow take(x, z))$	<>
Step 3	$\exists y.(course(y) \land \\ \forall x.(student(x) \Rightarrow take(x, y)))$	<>

