QR recap and argument raising

Patrick D. Elliott

December 5, 2022

Contents

1 Exercises (from Coppock and Champollion)

1.1 Composing quantifiers in object position

Produce a translation into the simply-typed lambda calculus for the following sentence:

(1) Beth speaks a European language.

Here are some translations to help you get started.

- Beth ⇒ $\mathbf{Beth} : E$
- speaks ⇒ $\mathbf{speaks} : E \to E \to T$
- European ⇒ $\lambda P_{ET} \,.\, \lambda x \,.\, \mathbf{european}(x) \land P(x) : (E \to T) \to E \to T$
- language ⇒ $\mathbf{language} : E \to T$
- a ⇒ $\lambda R_{ET} \,.\, \lambda S_{ET} \,.\, \exists x[P(x) \land Q(x)] : (E \to T) \to (E \to T) \to T$

Some important facts about quantifier raising.

- Traces are translated into *variables.*
- The moved expression introduces an *abstraction variable* into the LF, which triggers a special translation rule, *predicate abstraction.*

Predicate abstraction works as follows:

- γ is a syntax tree whose only two subtrees are x and β, where x is an abstraction variable.
- β is translated as β', an expression of type T
- Then translate γ as $\lambda x \,.\, \beta'$

Now, the exercise proper:

- Draw synax trees for the sentence (1), both before and after quantifier raising (assume that "a European language" undergoes QR). The syntax tree post-quantifier raising is called the "LF".
- Provide a translation for the sentence into the lambda calculus by compositionally translating each component part, and reducing the result using the reduction rules we've discussed.