

Homework # 4

due October 3

1 Reading

Please read Chapter 5 in your textbook.

2 Church Numerals

Do problem 5.2.4 from the book and then try it out using the `fulluntyped` implementation. What do you get for `powr three two`? Explain.

3 Proofs

Prove the following theorems in SASyLF:

1. Prove that multi-step full beta-reduction of ω where

$$\omega = (\lambda x \cdot x x)(\lambda x \cdot x x)$$

never reaches a “value,” (a lambda abstraction). To do this, you need to specify multi-step full beta-reduction. If you use the same technique (reflexive,transitive,inclusion) for multi-step evaluation as in Homework #3, you will require another lemma. It is easier if you define multi-step evaluation to be right recursive, as we did in the in-class proofs last week.

2. Prove that call-by-value evaluation is deterministic.

4 Graduate Students

For each of the following situations, give a pure lambda-calculus term that has the given properties:

1. diverges under normal evaluation and under call-by-value evaluation, but not under call-by-name
2. diverges under normal evaluation but not under call-by-value
3. diverges under call-by-value but not under normal evaluation

You may use ω defined above.

Call-by-need is used in lazy languages such as Haskell. Explain how it differs from the three: call-by-value, call-by-name and normal evaluation. Explain which of the three diverges on exactly the same terms as call-by-need, while still not being the same. Explain on paper.