Types and Programming Languages, Exercise 1

Philip Wadler


TAPL refers to *Types and Programming Languages* by Benjamin Pierce.

1. TAPL, Exercise 3.5.18

2. TAPL, Exercise 3.5.17

3. Write an implementation of untyped call-by-value lambda calculus based on the following principles:

   - use a big-step semantics;
   - use an environment rather than substitution at each step;
   - use variable names rather than de Bruijn indices;
   - include booleans and naturals as base types, with the operations given in the text (boolean constants, conditionals, zero, successor, predecessor, test for zero).

Base your on programs `arith` and `untyped` from Pierce’s implementation, found here:

[http://www.cis.upenn.edu/~bcpierce/tapl/resources.html#checkers](http://www.cis.upenn.edu/~bcpierce/tapl/resources.html#checkers)