CS 520 - Principles of Programming Languages
Solutions for Problem Set 5

Jue Wang

October 15, 2004

Problem 1: Exercise 13.1.3

Solution in [Pierce, page 513].

Problem 2: Exercise 13.5.2

Solution in [Pierce, page 515].

Problem 3:

Part 1: Exercise 13.5.8

Solution in [Pierce, page 516].

Part 2:

Solution on the next page.

Problem 4:

The following is a short program that takes two reference variables x and y of type Ref Nat and halts iff x and y are aliases of each other. This solution assumes the availability of the eq operator which was defined for problem 1 of problem set 3.

```
    req =  λf:Ref Nat → Ref Nat → Bool.
              λx:Ref Nat.
                λy:Ref Nat.
                  if eq !x !y then
                    y := succ !y;
                    if eq !x !y then true else f x y
                  else
                    f x y
▶ req :  (Ref Nat → Ref Nat → Bool) → Ref Nat → Ref Nat → Bool

   refEq = fix req

▶ refEq :  Ref Nat → Ref Nat → Bool
```

Problem 3, Part 2:

The following is the step by step evaluation of `(fact 3)`.

```
    (fact 3)
→   if iszero 3 then 1 else times 3 ((!fact_ref)(pred 3))
→   if false then 1 else times 3 ((!fact_ref)(pred 3))
→   times 3 ((!fact_ref)(pred 3))
→   times 3 ((λn:Nat.
                  if iszero n then 1 else times n ((!fact_ref)(pred n)))
               (pred 3))
→   times 3 ((λn:Nat.
                  if iszero n then 1 else times n ((!fact_ref)(pred n)))
               2)
→   times 3 (if iszero 2 then 1 else times 2 ((!fact_ref)(pred 2)))
→   times 3 (if false then 1 else times 2 ((!fact_ref)(pred 2)))
→   times 3 (times 2 ((!fact_ref)(pred 2)))
→   times 3 (times 2
                 ((λn:Nat.
                    if iszero n then 1 else times n ((!fact_ref)(pred n)))
                   (pred 2)))
→   times 3 (times 2
                 ((λn:Nat.
                    if iszero n then 1 else times n ((!fact_ref)(pred n)))
                   1))
→   times 3 (times 2 (if iszero 1 then 1 else times 1 ((!fact_ref)(pred 1))))
→   times 3 (times 2 (if false then 1 else times 1 ((!fact_ref)(pred 1))))
→   times 3 (times 2 (times 1 ((!fact_ref)(pred 1))))
→   times 3 (times 2 (times 1
                         ((λn:Nat.
                            if iszero n then 1 else times n ((!fact_ref)(pred n)))
                           (pred 1))))
→   times 3 (times 2 (times 1
                         ((λn:Nat.
                            if iszero n then 1 else times n ((!fact_ref)(pred n)))
                           0)))
→   times 3 (times 2 (times 1 (if iszero 0 then 1 else times 0 ((!fact_ref)(pred 0)))))
→   times 3 (times 2 (times 1 (if true then 1 else times 0 ((!fact_ref)(pred 0)))))
→   times 3 (times 2 (times 1 1))
→   times 3 (times 2 1)
→   times 3 2
→   6
```