CIS 705 — Programming Languages — Spring 2010

Assignment 4

Due by 2:30 p.m. on Tuesday, April 6

The context for this assignment is Chapters 5–7 of *TAPL*.

Exercise 1 (40 Points)

Summing a Range of Natural Numbers in Untyped Lambda Calculus

The goal of this exercise is to find a closed untyped lambda calculus term **sumRange** such that, for all closed terms t and t' and natural numbers n and m, if t converges to a closed value representing n, and t' converges to a closed value representing m, then

$$\textbf{sumRange}\ t\ t'$$

converges to a closed value representing the sum of

$$\{\, i \in \mathbb{N} \mid n \leq i \text{ and } i \leq m \,\}$$

(when $n > m$, this sum is 0).

Your answer should reside in a file named `sum-range-untyped.txt`. To test that your term works correctly on natural numbers n and m, copy your term to a new file, parenthesize it, and follow it with

c_n
c_m
```
(lambda x. lambda y. y x)
(lambda x. x);
```

Then give the untyped lambda calculus interpreter the new file as its input. The output should look like

```
(lambda y.
   y
   (lambda y'.
      y'
      (lambda y''.
         y''
         ... (lambda x.x) ...)))
```

where the number of occurrences of "`lambda y`"—with some number of primes on `y`—is the sum of $\{ i \in \mathbb{N} \mid n \leq i \text{ and } i \leq m \}$.

The Linux/Mac OS X shell script `sum-range-test` automates the process of creating the input to the untyped lambda calculus interpreter. Running

`./sum-range-test sum-range-untyped.txt` n m

puts the text described above in the file `sum-range-test-output.txt`. For example, running

```
./sum-range-test sum-range-untyped.txt 1 3
```

will put the text

```
( /* beginning of sumRange code */
  ...
/* end of sumRange code */ )

(lambda s. lambda z. s z) /* 1 */
(lambda s. lambda z. s(s(s z))) /* 3 */
(lambda x. lambda y. y x)
(lambda x. x);
```

in `sum-range-test-output.txt`. Giving this file to the untyped lambda calculus interpreter should produce the output

```
(lambda y.
   y
   (lambda y'.
      y'
      (lambda y''.
         y''
         (lambda y'''.
            y'''
            (lambda y''''.
               y'''' (lambda y'''''. y''''' (lambda x.x)))))))
```

because $1 + 2 + 3 = 6$.

Try to make your lambda calculus term as easy to understand as possible. Use comments, which begin with "`/*`" and end with "`*/`", to explain what you are doing. Format your term carefully. Although your term should run to completion on small inputs, you don't have to be concerned with its efficiency.

Submission

Your untyped lambda calculus term should begin with a comment containing your name. It should reside in an ordinary ASCII file, `sum-range-untyped.txt`. Submit your file by emailing it to me. I will acknowledge receiving it. Make sure that you retain an electronic copy of your file.

Exercise 2 (60 Points)

Proof of Correctness of sumRange

Prove that your answer to Exercise 1 is correct.

Submission

Submit your solution to this exercise on paper, not electronically.