

## Assignment 4

Due by 2:30 p.m. on Thursday, April 2

The context for this assignment is Chapters 5–7 of *TAPL* and Chapter 2 of *DAWOC*.

### Exercise 1 (60 Points)

The goal of this exercise is to prove the results that we called Propositions D–G in class, when discussing Chapter 5 of *TAPL*. You should prove these propositions in sequence, and you may use the results about  $\rightarrow$ ,  $\rightarrow^n$  and  $\rightarrow^*$  that were previously established.

**Proposition D (15 Points)** For all closed terms  $t_1$  and  $t_2$ , if  $t_1 t_2$  converges, then  $t_1$ ,  $t_2$  and  $\overline{t_1 t_2}$  converge and  $\overline{t_1 t_2} = \overline{\overline{t_1} \overline{t_2}}$ .

**Proposition E (15 Points)** For all closed terms  $t_1$  and  $t_2$ , if  $t_1$ ,  $t_2$  and  $\overline{t_1} \overline{t_2}$  converge, then  $t_1 t_2$  converges and  $\overline{t_1 t_2} = \overline{\overline{t_1} \overline{t_2}}$ .

**Proposition F (15 Points)** For all  $n \geq 2$  and closed terms  $t_1, \dots, t_n$ , if  $t_1 \dots t_n$  converges, then  $t_1, \dots, t_n$  and  $\overline{t_1} \dots \overline{t_n}$  converge and  $\overline{t_1 \dots t_n} = \overline{\overline{t_1} \dots \overline{t_n}}$ .

**Proposition G (15 Points)** For all  $n \geq 2$  and closed terms  $t_1, \dots, t_n$ , if  $t_1, \dots, t_n$  and  $\overline{t_1} \dots \overline{t_n}$  converge, then  $t_1 \dots t_n$  converges and  $\overline{t_1 \dots t_n} = \overline{\overline{t_1} \dots \overline{t_n}}$ .

### Submission

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

### Exercise 2 (40 Points)

The goal of this assignment is to translate the OCaml expression

```
let rec gcd n m =
  if m = 0
  then n
  else gcd m (n mod m)
in gcd
```

into the untyped lambda calculus. This expression has type  $\text{int} \rightarrow \text{int} \rightarrow \text{int}$  and is the well-known greatest common divisor function. Although its type involves integers, it is only

intended to be called with natural numbers. Of course, the untyped lambda calculus term that you translate it into won't be typed.

You should write a closed lambda calculus term that, when applied to closed values representing natural numbers  $n$  and  $m$ , converges to a closed value representing the greatest common divisor of  $n$  and  $m$ .

Your answer should reside in a file named `gcd-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 greatest common divisor of  $n$  and  $m$ .

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

```
gcd-test gcd-untyped.txt  $n$   $m$ 
```

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

```
gcd-test gcd-untyped.txt 4 6
```

will put the text

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

(lambda s. lambda z. s(s(s(s z)))) /* 4 */
(lambda s. lambda z. s(s(s(s(s(s z)))))) /* 6 */
(lambda x. lambda y. y x)
(lambda x. x);
```

in `gcd-test-output`. Giving this file to the untyped lambda calculus interpreter should produce the output (formatted slightly differently)

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

because the greatest common divisor of 4 and 6 is 2.

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, `gcd-untyped.txt`. Submit your program by emailing it to me. I will acknowledge receiving it. Make sure that you retain an electronic copy of your program.