CIS 500 — Software Foundations

Homework Assignment 4

Untyped lambda-calculus

Due: Monday, October 2, 2006, by noon

Instructions:

- Submit your solutions in PDF (or, if you must, ascii) format as `hw3`, for example, using the command:

```
~cis500/bin/cis500submit hw3 hw3.pdf
```

1 Exercise Show a single step of reduction for each of the following lambda-terms, or write "normal form" if a term cannot make a reduction step. For example, for `(λx. x) (λy. y)` you would write `(λy. y)`, while for `(λx. x)` you would write "normal form."

- `(λx. x (λy. x y)) (λx. x y x)`
- `(λx. x x) (λx. x x x)`
- `(λx. x x) (λx. x) (λx. x x)`
- `(λx. (λy. y y) x)`

2 Exercise Using the derivation tree shown in TAPL just before Exercise 3.5.14 as a model, draw a complete derivation tree for the following single-step reduction:

$$\texttt{(λx. x x) (λy. y) (λz. z z)} \longrightarrow \texttt{(λy. y) (λy. y) (λz. z z)}$$

If you are using Latex, feel free to use a simple `verbatim` environment to typeset your solution — no need for anything fancy.

3 Exercise For each of the terms in exercise 1, write its normal form, if any. For terms that have no normal form, write "diverges."

4 Exercise TAPL exercise 5.2.7.

Again, if you are using Latex, use a simple `verbatim` environment to typeset your solution.

Also, for this and the following exercise, you can test your programs on Eniac (and other SEAS machines) by typing:

```
~cis500/bin/fulluntyped YOURFILENAME
```

(Of course, you can also use this checker to verify your answers to the problems above, but we recommend that you work them by hand first.)

The file `hw04.f` gives an example of the syntax expected by the `fulluntyped` tool. A link to this file can be found on the course homework page—or just get it straight from here:

```
http://www.seas.upenn.edu/~cis500/hw/hw04.f
```

5 Exercise TAPL exercise 5.3.6.

6 Exercise Use the fixed point operator `fix` defined in TAPL (or the `Z` operator defined in the lecture notes), together with the church numerals and booleans, to implement a function that tests whether a given church numeral is odd. For example, `odd c0` should evaluate to `fls`, while `odd c1` and `odd c3` should both evaluate to `tru`.

7 Exercise [Required for groups containing at least one PhD student; optional for others.] A lambda-term `t` is said to be *linear* if, for every subterm of `t` of the form `λx. s`, the bound variable `x` occurs exactly once in the body `s`. For example, the terms

```
λx. x
λx. (λy. y) x
λx. (λy. x y)
```

are all linear, while

```
λx. x x
λx. (λy. y)
```

are not.

Prove that every linear term has a normal form.

8 Debriefing

1. Approximately how many hours (per person, on average) did you spend on this assignment?
2. Would you rate it as easy, moderate, or difficult?
3. How deeply do you feel you understand the material it covers (0%–100%)?
4. Any other comments?