CIS 500 Software Foundations (Fall 2002)

Homework Assignment 11

Implementing Subtyping

Due: Thursday, November 21, 2002, by 4PM

1 Exercise 16.2.5 in TAPL.

2 Exercise In

```
http://www.seas.upenn.edu/~cis500/hw/errorexercise
http://www.seas.upenn.edu/~cis500/hw/errorexercise.tar
```

you will find an OCaml implementation of a typechecker and interpreter for a simply typed lambda-calculus with subtyping, records, and simple exceptions.

Several parts of the file `core.ml` have been omitted and marked with a comment like

```
(* ############ Fill in ########### *)
```

(In some cases, the omitted text is replaced by `assert false` so that the file will still compile.)

Fill in all these parts.

You can test your code by building an executable file `f` and then typing `./f test`. The output should be:

```
<fun> : Top -> Top
<fun> : Top
<fun> : Top -> Top
<fun> : Top
<fun> : Bot -> Bot
<fun> : Bot -> Bot
{x=true, y=false, a=false} : {x:Top, y:Bool}
<fun> : Bool -> Bot
true : Bool
error : Bool
error : Bot
error : Bool
error : Bool
false : Bool
error : {x:Bool, y:{a:Bot}, z:Bool}
```

3 Debriefing

1. How many hours did you spend on this assignment?
2. Would you rate it as easy, moderate, or difficult?
3. How much did you rely on our solutions while developing yours? (Completely / substantially / moderately / just for verification / not at all?)
4. Did you work on it mostly alone, or mostly with other people?
5. How deeply do you feel you understand the material it covers (0%–100%)?
6. Any other comments?

The solution to exercise (1) is in the back of TAPL. The solution to exercise (2) can be found here:

```
http://www.seas.upenn.edu/~cis500/hw/errorsolution
http://www.seas.upenn.edu/~cis500/hw/errorsolution.tar
```