Homework Assignment 4 (HW4)
Posted Friday, April 4, 2014
Due Monday, April 14, 2014

Problem 1 (15pts). [Modified from Sethi]. Let

\[ S = \lambda xyz. xz(yz) \quad \text{and} \quad I = \lambda x.x \]

Reduce \( SIII \) to the normal form \( \lambda x.x \) using applicative order reduction and normal order reduction.

Problem 2 (25pts). [Modified from Sethi]. Verify the following equality:

\[ \text{twice} (\text{twice}) f x =_{\alpha\beta} f (f (f (f x))), \] where \( \text{twice} \) is \( \lambda f x.f (f x) \)

a. Using applicative order reduction
b. Using normal order reduction

Problem 3 (15pts). [From Sethi]. Show that the term \( ZZ \) where \( Z \) is \( \lambda z. \lambda x. x(zz) \) satisfies the requirement for fixed-point combinators that \( ZZM =_{\beta} M(ZZM) \).

Problem 3 (10pts). [Modified from Scott]. In the following code, which of the variables will a compiler consider to have compatible types under structural equivalence? Under strict name equivalence? Under loose name equivalence?

```plaintext
type T = array [1..10] of integer
type S = T
A : T
B : T
C : S
D : array [1..10] of integer
```

Problem 4 (20pts). [Modified from Scott]. Explain the meaning of the following C declarations. Draw the type trees as we did in class.

```plaintext
double *a[n];
double (*b)[n];
double (*c[n])();
double (*d())[n];
```

Problem 5 (15pts). [Modified from Scott]. Consider the following declaration in C:

```plaintext
double (*bar(int, double(*)(double,double[]))) (double);
```
Describe in English the type of \( \text{bar} \). Draw the type tree.

How about

```plaintext
double (**bar)(int, double(*)(double,double[]))(double);
```
Describe and draw the type tree. Is this a valid declaration in C? Explain your answer.