

Programming Languages Homework 4

Tate Larsen

2011-11-22 Tue

Problem 1

a

- $S = \lambda xyz.(x\ z)(y\ z)$
- $I = \lambda x.x$
- $SIH = \lambda xyz.(x\ z)(y\ z) (\lambda x.x) (\lambda x.x) (\lambda x.x) = (\lambda x.x\ \lambda x.x) (\lambda x.x\ \lambda x.x) = \lambda x.x\ \lambda x.x = \lambda x.x = I$

b

- $\text{twice} = \lambda fx.f(f\ x)$
- $\text{twice} (\text{twice})\ f\ x = \text{twice} (\text{twice}\ f)\ x = \text{twice} = \text{twice}\ f\ (\text{twice}\ f\ x) = \text{twice}\ f\ f(f(x)) = f(f(f(x)))$

Problem 2

- $Z = \lambda z. \lambda x.x(z\ z\ x)$
- $ZZM = ((\lambda z. \lambda x.x(z\ z\ x))(\lambda z. \lambda x.x(z\ z\ x)))M = (\lambda x.x\ (\lambda z. \lambda x.x(z\ z\ x)))(\lambda z. \lambda x.x(z\ z\ x))\ x)M = M((\lambda z. \lambda x.x(z\ z\ x))(\lambda z. \lambda x.x(z\ z\ x))M) = M(ZZM)$

Problem 3

Structural Equivalence

- $A = B = C = D$

Strict Name Equivalence

- A and B are compatible

Loose Name Equivalence

- A, B, and C are compatible

Problem 4

`double *a[n];`

- Array of n-many pointers to doubles

`double (*b)[n];`

- Pointer to array of n-many doubles

`double (*c[n])();`

- Array of n-many pointers to functions which return doubles

`double (*d())[n];`

- Function which returns pointer to array of n-many doubles

Problem 5

- foo is a function that takes in 1) a pointer to a function that takes in a double and array of doubles and returns a double and 2) a double and returns a pointer to a function that takes a double and “...” and returns a double.

Problem 6

- Assume: $s_i = \text{sizeof}(\text{int})$
- Address of `a[j][k,l]` will be $\text{addr}(a) + j * 25 * s_i + k * 5 * s_i + l * s_i$