CS 465: Language, Logic and Computation—Lambda Calculus Homework

Fritz Ruehr—Spring 2015

This homework is based on the material presented in lecture; you should refer to on-line hand-outs and your own lecture notes for definitions, etc. Please ask for help if you have any questions.

1. *Syntactic abbreviations*

Convert the following lambda terms into fully parenthesized form, with all multi-variable abstractions expanded. Let's agree that "fully parenthesized" will mean that all applications and abstractions should be parenthesized, except for the outermost term (i.e., at top level).

- λxyz. f x (f y y) (f z)
- λxy. f (λx. x) (λgf. g f x)

Perform the "opposite operation" on the following terms; i.e., use standard conventions on parenthesization and multiple abstractions to write them in a *minimal* form.

- λa. (λb. (a ((b b) a))
- (((λp. (λx. (p x)) (p a)) (λq.q)) (λy.b)) a

2. *Variables and binding*

Convert the following terms so that no free or bound variables clash, i.e., so that all variables are distinct (respect the existing variable bindings: you may change the variables, but preserve the meaning).

- (λx. (λxy. x (y x)) y) (λy. (λy. y x) y x)
- b a (λab. a (λa. b) b) a (λb. b a)

3. *Substitution and reduction*

Reduce the following lambda and combinator terms to normal form, using normal-order reduction, or argue that they have no normal form. Show intermediate β-reduction steps and, if necessary, variable renamings to avoid capture.

- (λfgx. f (g x)) (λy.y) (λfx. f (f x)) x
- (λfx. f (f x)) (λy. c (y y)) b
- **S I I (S I I)**