Quiz 3

Date: Jul 16, 2020

- [8pts] Q1 - Lambda Operations
  - [2pts] Q1.1 - Explicit Parenthesis
  - [3pts] Q1.2 - Alpha Equivalence
  - [3pts] Q1.3 - Beta Reductions
- [6pts] Q2 - Beta Normal Form
  - [3pts] Q2.1 -
  - [3pts] Q2.2 -
- [6pts] Q3 - Encoding Exercises

[2pts] Q1.1 Explicit parenthesis

Which of the following options are equivalent to `\x. x \f. \a. x f f f a`?

`[ ] ` `(\x. x) (\f. (\a. (((x f) f) f) a))`

`[X] ` `(\x. (x (\f. (\a. (((x f) f) f) a))))`

`[ ] ` `(\x. x) (\f. (\a. (x (f (f (f a))))))`

`[ ] ` `(\x. (x (\f. (\a. (x (f (f (f a)))))))`

[3pts] Q1.2 Alpha Equivalence

Which of the following are alpha equivalent `\a. \b. a b c`?

`[ ] ` `\a. \b. b a c`

`[ ] ` `\b. \a. a b c`

`[X] ` `\b. \a. b a c`

`[X] ` `\x. \y. x y c`

[3pts] Q1.3 Beta reductions

Which of the following is the result of *any number* of alpha conversions and a *single* beta reduction of `(\a. (\x. x a) (\a. x a)) (\x. \y. y)`?

`[ ] ` `(\a. (\x. x a) (\a. x a)) (\x. \y. y)`

`[X] ` `(\a. (\a. x a) a) (\x. \y. y)`

`[X] ` `(\x. x (\x. \y. y)) (\y. x y)`

`[ ] ` `x (\x. \y. y)`

[6pts] Q2 Beta Normal Form

Reduce the following to beta normal form. For ease of grading and assigning partial credit, please write each alpha conversion and beta reduction on a new line.

[3pts] 2.1

`(\a. \b. a a b) (\a. \b. a) (\a. \b. a)`

Soln:

`(\a. \b. a) (\a. \b. a) (\a. \b. a)`

beta `(\b. (\a. \b. a)) (\a. \b. a)`

beta `(\a. \b. a)`

[3pts] 2.2

`(\a. \b. b) ((\x. x x) (\x. x x)) x`

Soln:

`(\a. \b. b) ((\x. x x) (\x. x x)) x`

beta `(\b. b)  x`

beta `x`

[6pts] Q3. Exercise Encoding

Using church encodings simplify the following expression.

If possible, the result should be written as a Church Encoding. e.g. `(\f. \x. f (f x))` is `2`

Simplify `1 1 2` to the simplest form.

`1 1 2`

Soln:

decode `(\f. \x. f x) (\f. \x. f x) (\f. \x. f (f x))`

alpha `(\f. \x. f x) (\f. \a. f a) (\f. \x. f (f x))`

beta `(\x. (\f. \a. f a) x)  (\f. \x. f (f x))`

beta `(\f. \a. f a) (\f. \x. f (f x))`

beta `(\a. (\f. \x. f (f x)) a)`

beta `(\a. (\x. a (a x)))`

alpha `(\f. (\x. f (f x)))`

encode `2`