

ASSIGNMENT 3 — LAMBDA CALCULUS

COMP 3010 — ORGANIZATION OF PROGRAMMING LANGUAGES

1. SYNTAX

Remember that the abstract syntax of the λ -calculus is defined by the grammar

$$M ::= x \mid M \ M \mid \lambda x.M$$

where x , y , and z stand for *identifier names* (a.k.a. *variables*) used inside expressions in the language of the λ -calculus.

Ambiguities in the grammar, where multiple juxtapositions and λ s are mixed, are resolved by these associativity and precedence rules:

$$\begin{aligned} M_1 \ M_2 \ \dots \ M_n &= ((M_1 \ M_2) \ \dots) \ M_n & M_1 \ \lambda x.M_2 &= M_1 \ (\lambda x.M_2) \\ \lambda x.M_1 \ M_2 \ \dots \ M_n &= \lambda x.(M_1 \ M_2 \ \dots \ M_n) & \lambda x_1.\lambda x_2.M &= \lambda x_1.(\lambda x_2.M) \end{aligned}$$

In other words, application (written as juxtaposition $M_1 \ M_2$) associates to the *left*, abstraction (written as $\lambda x.M$) associates to the *right*, a λ to the right of an application (a λ argument $M_1 \ (\lambda x.M_2)$) has a *higher precedence* than that application, and an application to the right of a λ (an applied function body $\lambda x.(M_1 \ M_2)$) has a *higher precedence* than the λ .

Exercise 1 (Multiple Choice). Which of the following, fully-parenthesized λ -calculus expressions stand for the same syntax tree as $y \ \lambda x.x \ y \ z$?

- (a) $y \ (\lambda x.x) \ (y \ z)$
- (b) $((y \ (\lambda x.x)) \ y) \ z$
- (c) $y \ (\lambda x.(x \ (y \ z)))$
- (d) $y \ (\lambda x.((x \ y) \ z))$

Exercise 2 (Short Answer). Use the above rules of precedence/associativity to draw the syntax tree represented by the fully-parenthesized version of $\lambda x.\lambda y.y \ z \ x$

2. ALPHA

Exercise 3 (Short Answer). Remember that the set of *free variables* of an expression M , written as $\text{FV}(M)$, is *inductively defined* on the *syntax* of M like so:

$$\begin{aligned} \text{FV}(x) &= \{x\} \\ \text{FV}(M_1 \ M_2) &= \text{FV}(M_1) \cup \text{FV}(M_2) \\ \text{FV}(\lambda x.M) &= \text{FV}(M) - \{x\} \end{aligned}$$

Give the set of *free variables* found in the expression $(\lambda x.\lambda y.y \ (\lambda z.x)) \ (\lambda x.z)$:

$$\text{FV}((\lambda x.\lambda y.y \ (\lambda z.x)) \ (\lambda x.z)) = ?$$

Exercise 4 (This or That). Remember that the operation which *renames* x to y in M , written $[x/y]M$, is *inductively defined* by transforming the *syntax* of M as:

$$\begin{aligned} [y/x]x &= x \\ [y/x]z &= z & (\text{if } z \neq x) \\ [y/x](M_1 \ M_2) &= ([y/x]M_1) ([y/x]M_2) \\ [y/x](\lambda x.M) &= \lambda x.M \\ [y/x](\lambda z.M) &= \lambda z.([y/x]M) & (\text{if } z \neq x) \end{aligned}$$

The law of α -equivalence is defined in terms of renaming as:

$$\lambda x.M =_\alpha \lambda y.([y/x]M)$$

Determine which of the following are valid renamings. In other words, say whether or not each pair of λ -calculus expressions are equal according to α -equivalence.

- (1) $\lambda x.(x \ x) =_\alpha \lambda y.(y \ y)?$
- (2) $\lambda x.(\lambda x.x) =_\alpha \lambda y.(\lambda z.z)?$
- (3) $\lambda x.(x \ y) =_\alpha \lambda y.(y \ y)?$
- (4) $(\lambda x.x) \ x =_\alpha (\lambda y.y) \ x?$
- (5) $(\lambda x.x) \ x =_\alpha (\lambda y.y) \ y?$

3. BETA

Exercise 5 (Multiple Choice). Remember that the operation which *substitutes* M' for x in M , written $[M'/x]M$, is *inductively defined* by transforming the *syntax* of M like so:

$$\begin{aligned} [M'/x]x &= M' \\ [M'/x]z &= z & (\text{if } z \neq x) \\ [M'/x](M_1 \ M_2) &= ([M'/x]M_1) ([M'/x]M_2) \\ [M'/x](\lambda x.M) &= \lambda x.M \\ [M'/x](\lambda z.M) &= \lambda z.([M'/x]M) & (\text{if } z \neq x \text{ and } z \notin \text{FV}(M')) \end{aligned}$$

What is the result of the substitution $[(\lambda x.x \ x)/y](y \ (\lambda z.y \ z))$?

- (1) $[(\lambda x.x \ x)/y](y \ (\lambda z.y \ z)) = (\lambda x.x \ x) \ (\lambda z.y \ z)$
- (2) $[(\lambda x.x \ x)/y](y \ (\lambda z.y \ z)) = y \ (\lambda z.(\lambda x.x \ x) \ z)$
- (3) $[(\lambda x.x \ x)/y](y \ (\lambda z.y \ z)) = (\lambda x.x \ x) \ (\lambda z.(\lambda x.x \ x) \ z)$
- (4) $[(\lambda x.x \ x)/y](y \ (\lambda z.y \ z)) = (\lambda x.x \ x) \ (\lambda z.(\lambda x.x \ x) \ (\lambda x.x \ x))$

Exercise 6 (Multiple Choice). Remember that the law of β -reduction is defined in terms of substitution as:

$$(\lambda x.M) \ M' \mapsto_\beta [M'/x]M$$

and the *call-by-name operational semantics* for the λ -calculus is defined by applying the above β -reduction rule inside of these evaluation contexts E defined like so:

$$E ::= \square \mid E \ M \qquad \frac{M \mapsto_\beta M'}{E[M] \mapsto_\beta E[M']}$$

Which of the following is the result of evaluating the expression

$$(\lambda x.\lambda y.\lambda z.y \ x \ z) \ (\lambda x.x) \ (\lambda x.\lambda y.x) \ z$$

according to the call-by-name operational semantics (as shown above)? In other words, which of the following do you get by applying β -reduction (as shown above) as many times as possible in a sequence like so:

$$(\lambda x. \lambda y. \lambda z. y \ x \ z) (\lambda x. x) (\lambda x. \lambda y. x) \ z \mapsto M_1 \mapsto M_2 \mapsto \dots \mapsto \text{answer}$$

- (a) $(\lambda x. x)$
- (b) $(\lambda x. \lambda y. x)$
- (c) $(\lambda y. z)$
- (d) z

Exercise 7 (Short Answer). What happens when you evaluate the λ -calculus expression $(\lambda x. x \ x) (\lambda x. x \ x)$? by repeatedly applying β -reduction? If β -reduction eventually stops, what final expression is the result? If you think β -reduction does not stop, explain why not?

4. ETA

Exercise 8 (This or That). Remember that the law of η -reduction is defined in terms of free variables like so:

$$(\lambda x. (M \ x)) \rightarrow_{\eta} M \qquad (\text{if } x \notin \text{FV}(M))$$

Say which of the following are correct η -reductions (according to the above rule) and which are not.

- (1) $\lambda x. x \rightarrow_{\eta} x$
- (2) $\lambda y. (x \ y) \rightarrow_{\eta} x$
- (3) $\lambda x. (x \ x) \rightarrow_{\eta} x$
- (4) $\lambda x. ((\lambda y. y) \ x) \rightarrow_{\eta} \lambda y. y$
- (5) $\lambda x. ((y \ x) \ x) \rightarrow_{\eta} y \ x$