CSE340 Fall 2018 Homework 3

Due Tuesday October 2 2018 by 11:59 PM on Blackboard

Note: your answers can be handwritten, but you need to have clear handwriting. If we cannot read it we cannot grade it!

Problem 1 (binding). For each of the following determine for each variable x the $\lambda x.$ it is bound to. I have numbered the variables and the abstractions. The variables are numbered using Arabic numerals and the abstractions are numbered using roman numerals. If variables 4 and 7 are bound to abstraction I, your answer should be of the form I → 4, 7 to indicate that abstraction I has variables 4, and 7 bound by it.

Example $(\lambda x.\ x\ x\ \lambda x.\ x\ )\ x$

I 1 2 II 3 4

Answer

I → 1, 2
II → 3

---

a. $x\ \ \lambda x.\ (\ \lambda x.\ \ \lambda x.\ \ x\ \ \ x\ )\ x$

1 I II III 2 3 4

Answer

I → 4
II →
III → 2, 3

b. $\lambda x.\ \ x\ \ \lambda x.\ \ x\ \ \ x$

I 1 II 2 3

Answer

I → 1
II → 2, 3

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

I II 1 III 2 3

Answer

I → 1
II →
III → 2, 3

f. $(\ \lambda x.\ \ (\ \lambda x.\ \ (\ \lambda x.\ \ x\ \ )\ \ x\ \ (\ \lambda x.\ \ x\ \ )\ \ x\ )\ \ (\ \lambda x.\ \ x\ \ x\ \ )\ \ (\ \lambda x.\ \ x\ \ )\ \ \ )\ \ x$

I II III 1 2 IV 3 4 V 5 6 VI 7 8

Answer

I →
II → 2 , 4
III → 1
IV → 3
V → 5 , 6
VI → 7

Problem 2 (reducible expressions). In this problem you are asked to determine the reducible expressions in each of the following lambda expressions. The format of the answer is shown in the examples below

a. x λx. (λx. λx. x x) x

(λx. t) t’

b. λx. x λx. x x NONE

c. λx. (λy. x) λx. x x

(λx. t) t’

d. λx. λx. x x x NONE

e. (λx. x x) (λx. x) x

(λx. t) t’

f. (λx. (λx. (λx. x) x (λx. x) x) (λx. x x) (λx. x)) x

(λx. t) t’

(λx. t) t’

(λx. t) t’

Examples to show the format of the answer

Example 1: for the expressions

λx. x x (λx. x) x λx. x x

Your answer should be that there are no reducible expressions.

Example 2: for the expression

(λx. (λx. x) x) λx. x x

Your answer should be

(λx. (λx. x) x) λx. x x

(λx. t) t’

(λx. (λx. x) x) λx. x x

(λx. t) t’

Problem 3 (Beta Reductions). For each of the following expressions, give the resulting expression after executing a beta reduction of the highlighted redex (with renaming **only if needed**. You should not do renaming if it is not needed). Examples are shown below.

a. $(\underbrace{\lambda x.}_{(\lambda x.}\ \underbrace{x\ \ y}_{t}\,\underbrace{)}_{)}\ \underbrace{\lambda y.\ y\ y}_{t'} \xrightarrow{\beta} (\lambda y.\ y\ y)\ y$

b. $(\underbrace{\lambda x.}_{(\lambda x.}\ \underbrace{x\ (\lambda y.\ x\ \lambda y.\ \ y)\ x}_{t}\,\underbrace{)}_{)}\ \underbrace{\lambda x.\ y\ x}_{t'} \xrightarrow{\alpha} (\lambda x.\ x\ (\lambda w.\ x\ \lambda y.\ \ y)\ x)\ (\lambda x.\ y\ x)$

$\xrightarrow{\beta} (\lambda x.\ y\ x)\ (\lambda w.\ (\lambda x.\ y\ x)\ \lambda y.\ \ y)\ (\lambda x.\ y\ x)$

c. $(\underbrace{\lambda x.}_{(\lambda x.}\ \underbrace{x\ \ \ \lambda x.\ \lambda z.\ x\ x}_{t}\,\underbrace{)}_{)}\ \underbrace{\lambda x.\ y\ x}_{t'} \xrightarrow{\beta} (\lambda x.\ y\ x)\ \ \ \lambda x.\ \lambda z.\ x\ x$

d. $(\underbrace{\lambda x.}_{(\lambda x.}\ \underbrace{x\ y\ \ \lambda z.\ \lambda y.\ x}_{t}\,\underbrace{)}_{)}\ \underbrace{\lambda y.\ \lambda z.\ x\ y\ z}_{t'} \xrightarrow{\beta} (\lambda y.\ \lambda z.\ x\ y\ z)\ \ y\ \ \lambda z.\ \lambda y.\ (\lambda y.\ \lambda z.\ x\ y\ z)$

Example 1: for the expression

$(\underbrace{\lambda x.}_{(\lambda x.}\ \underbrace{(\lambda x.\ x\ )\ x}_{t}\,\underbrace{)}_{)}\ \underbrace{\lambda x.\ x\ x}_{t'}$

your answer should be

$(\lambda x.\ x\ )\ (\lambda x.\ x\ x\ )$ // beta reduction

Example 2: for the expression

$(\underbrace{\lambda x.}_{(\lambda x.}\ \underbrace{(\lambda x.\ x\ )\ \lambda y.\ y\ x}_{t}\,\underbrace{)}_{)}\ \underbrace{\lambda x.\ y\ x}_{t'}$

your answer should be

$(\lambda x.\ (\lambda x.\ x\ )\ \lambda w.\ w\ x\ )\ \lambda x.\ y\ x$ // renaming
$(\lambda x.\ x\ )\ \lambda w.\ w\ (\lambda x.\ y\ x\ )$ // beta reduction

Problem 4 (Call by Value). For each of the following, identify the redex that should be reduced first **according to call by value** and give the resulting expression after one beta reduction. If there is more than one possibility, you should show all possibilities.

a. $(\lambda x.\ (\lambda z.\ \lambda y.\ x)\ x\ y\ z)$ NONE

b. $x\ (\lambda x.\ x\ \ \lambda x.\ \lambda y.\ x\ x\ )\underbrace{(\ (\lambda x.\ y)\ x\ )}_{1}\underbrace{(\ (\lambda x.\ y)\ x\ )}_{2}$ either 1 or 2 can be reduced first

$\xrightarrow{\beta\,1}$ $x\ (\lambda x.\ x\ \ \lambda x.\ \lambda y.\ x\ x\ )(\ y\ )(\ (\lambda x.\ y)\ x\ )$

$\xrightarrow{\beta\,2}$ $x\ (\lambda x.\ x\ \ \lambda x.\ \lambda y.\ x\ x\ )(\ (\lambda x.\ y)\ x\ )(\ y\ )$

c. $\underbrace{(\ (\lambda x.\ x\ )(\underbrace{\ (\lambda x.\ y\ )\ x\ }_{1})\ )}_{3}\ \underbrace{(\ (\lambda x.\ y\ )\ x\ )}_{2}$

either 1 or 2 can be reduced first
3 cannot be reduced because its argument is not a value

$\xrightarrow{\beta\,1}$ $(\ (\lambda x.\ x\ )(\ y\ )\ )\ (\ (\lambda x.\ y\ )\ x\ )$

$\xrightarrow{\beta\,2}$ $(\ (\lambda x.\ x\ )(\ (\lambda x.\ y\ )\ x\ )\ )\ (\ (\ y\ )\ )$

d. $\underbrace{(\lambda x.\ \underbrace{(\ \lambda x.\ x\ )}_{2}\ x\ (\ \lambda x.\ x\ )\ x\ )\ (\lambda x.\ y)}_{1}\ x$

1 should be reduced first
2 is under an abstraction

$\xrightarrow{\beta\,1}$ $(\ \lambda x.\ x\ )\ (\lambda x.\ y)\ \ (\ \lambda x.\ x\ )\ (\lambda x.\ y)\ x$

Notation I use $\xrightarrow{\beta\,1}$ to indicate the result of beta reduction of redex 1

Problem 5 (Normal Order of Evaluation). For each of the following, identify the redex that should be reduced first **according to normal order evaluation** and give the resulting expression after one beta reduction.

a. $(\lambda x.\ x\ (\lambda z.\ \lambda y.\ x)\ x\ y)$ NONE

b. $\underbrace{(\lambda x.\ x\ \ \lambda x.\ \lambda y.\ x\ \ x\ \ )\ \underbrace{((\lambda x.\ y)\ x)}_{1}}_{3}\ \underbrace{((\lambda x.\ y)\ x)}_{2}$

3 is leftmost outermost so it is reduced first

$\xrightarrow{\beta\ 3}\ (((\lambda x.\ y)\ x)\ \lambda x.\ \lambda y.\ x\ \ x)\ ((\lambda x.\ y)\ x)$

c. $\lambda x.\ (\underbrace{(\lambda x.\ y\ )\ x}_{1})\ (\underbrace{(\lambda x.\ y)\ x}_{2})\ (\lambda x.\ y\ z)$

1 is leftmost outermost so it is reduced first

$\xrightarrow{\beta\ 1}\ \lambda x.\ y\ ((\lambda x.\ y)\ x)\ (\lambda x.\ y\ z)$

d. $\underbrace{(\lambda x.\ \underbrace{(\lambda x.\ x)\ x}_{1}\ (\lambda x.\ x)\ x)\ (\lambda x.\ y)}_{2}\ x$

2 is leftmost outermost so it is reduced first

$\xrightarrow{\beta\ 2}\ \mathbf{(}(\lambda x.\ x)\ (\lambda x.\ y)\ \ (\lambda x.\ x)\ (\lambda x.\ y)\ \ )\ x$

Problem 6 (Calculation with Lambda Expressions). For each of the following, write a lambda expression to calculate the function. You can use any of the functions I covered in class or in the slides, but you should not use recursion for any of the functions. For all questions the functions are for numbers represented as Church numerals.

---

a. ONETRUE = (𝜆a. 𝜆b. 𝜆c. ......) should calculate ONETRUE a b c defined as follows

1. ONETRU fls fls fls = fls
2. ONETRU fls fls tru = tru
3. ONETRU fls tru fls = tru
4. ONETRU fls tru tru = fls
5. ONETRU tru fls fls = tru
6. ONETRU tru fls tru = fls
7. ONETRU tru tru fls = fls
8. ONETRU tru tru tru = fls

AND = 𝜆a. 𝜆b. a b fls
NOT = 𝜆a. a fls tru
ONETRUE = 𝜆a. 𝜆b. 𝜆c. a (b fls (NOT c)) (b (NOT c) c)

Explanation. if we apply ONETRUE to three arguments x, y and z we get

x (y fls (NOT z) (y (NOT z) z)

- If x is tru, the result is (y fls (z fls tru))
  - if y is tru, the result is fls (x and y are try)
  - if y is fls, the result is NOT z, which is tru if z is fls and fls if z is tru
- if x is fls, the result is (y (NOT z) z)
  - if y is tru, the result is NOT Z
  - if y is also fls, the result is z

b. sum_products = (𝜆n. ......) should calculate 1*2 + 3*4 + 5*6 + n*(n+1)

note: for n*(n+1) to make sense in this definition, we need n to be odd. Or, we fix it and simply calculate up to (2n-1)*(2n). I will give the answer for the later.

init = pair 1 0
next = 𝜆p. pair ((succ (succ (fst p))) (plus (snd p) (times (fst p) (succ (fst p)))))

sum_products = (𝜆n. snd (n next init))

Explanation If we apply next to n and s we get

next (pair i s) = pair (i+2) (s + i*i+1)

Note 1. I write the expression in more familiar notation to explain what is happening
Note 2. If we want to get the sum up to the largest odd number less than or equal to n, we can first calculate n DIV 2 (see next problem) and apply it to next and init.

Problem 6 (Calculation with Lambda Expressions). For each of the following, write a lambda expression to calculate the function. You can use any of the functions I covered in class or in the slides, but you should not use recursion for any of the functions. For all questions the functions are for numbers represented as Church numerals.

---

c. DIV_3 **=** integer division by 3.

Examples DIV_3 2 = 0
DIV_3 5 = 1
DIV_3 6 = 2
DIV_3 7 = 2

e. (Bonus) Can you generalize this for integer division by n?

I give the general solution

init = pair 0 0
add_by_n = λn. λp. (equal (succ (fst p)) n)
(pair 0 (succ (snd p)))
(pair (succ (fst p)) (snd p))

div_m_n = λm. λn. snd (m (add_by_n n) init)