CSE340 FALL 2021 HOMEWORK 4
Due Friday November 12, 2021 by 11:59 PM

PLEASE READ THE FOLLOWING CAREFULLY

- Your answers must be typed.
- On Gradescope, you should submit the answers to separate question separately.
- Read carefully the required answer format. The required format will make it easier for you to answer and for the graders to grade. Answers that are not according to the required format will not be graded
- You do not need to show your work, but remember that this homework is very important for EXAM 2.

Problem 1 (Lambda Calculus) The goal of this problem is to give you practice with lambda calculus. Each part of this problem will have an expression that you are asked to evaluate or simplify as much as possible. The following are some examples

Example 1. plus2 = λn. succ (succ n)
what does the following evaluate to: 4 plus2 2
Answer. 10

Example 2. quad = λx. λy. λz. λw. pair (pair x y) (pair z w)
what does the following evaluate to: succ (fst (snd (quad 1 3 5 7)))
Answer. 6

We will use the following definitions in what follows

quad = λx. λy. λz. λw. pair (pair x y) (pair z w)
1st = λq. fst (fst q)
2nd = λq. snd (fst q)
3rd = λq. fst (snd q)
4th = λq. snd (snd q)
tri = λx. λy. λz. pair x (pair y z)
f0 = λp. pair **(**TIMES (fst p) (snd p)**)** **(**TIMES (fst p) (snd p)**)**
f1 = λq. quad (2nd q) (3rd q) (4th q) (1st q)
f2 = λt. tri (OR (fst t) **(**EQUAL (fst(snd t)) (snd (snd t))**)**) // first element
(TIMES (fst (snd t)) 2) // second element
(snd (snd t)) // third element

For each of the following, give the value that the expressions evaluates to

1. What is f0 (pair 1 2)?

   Answer. f0 (pair 1 2)

   = (λp. pair **(**TIMES (fst p) (snd p)**)** **(**TIMES (fst p) (snd p)**)**)(pair 1 2)

   = pair **(**TIMES (fst (pair 1 2)) (snd (pair 1 2))**)** **(**TIMES (fst (pair 1 2)) (snd (pair 1 2))**)**

= pair **(**TIMES 1 2**)** **(**TIMES 1 2**)**

= pair 2 2

2. What is f0 (f0 (pair 1 2))?

   Answer. f0 (f0 (pair 1 2))

   = f0 (pair 2 2)

   = (λp. pair **(**TIMES (fst p) (snd p)**)** **(**TIMES (fst p) (snd p)**)**)(pair 2 2)

   = pair **(**TIMES (fst (pair 2 2)) (snd (pair 2 2))**)** **(**TIMES (fst (pair 2 2)) (snd (pair 2 2))**)**

   = pair **(**TIMES 2 2**)** **(**TIMES 2 2**)**

   = pair 4 4

3. What is f0 (f0 (f0 (pair 1 2)))?

   Answer. f0 (f0 (f0 (pair 1 2)))

   = f0 (pair 4 4)

   = (λp. pair **(**TIMES (fst p) (snd p)**)** **(**TIMES (fst p) (snd p)**)**)(pair 4 4)

   = pair **(**TIMES (fst (pair 4 4)) (snd (pair 4 4))**)** **(**TIMES (fst (pair 4 4)) (snd (pair 4 4))**)**

   = pair **(**TIMES 4 4**)** **(**TIMES 4 4**)**

   = pair 16 16

4. What does the function λn. fst (n f0 (pair 1 2)) calculate for a Church's numeral n? Give a *compact* description.
   Answer. The function returns 2^x where $x = 2^{n-1}$

5. what is f1 (quad 1 1 1 1)?

   Answer. f1 (quad 1 1 1 1)

   = (λq. quad (2nd q) (3rd q) (4th q) (1st q))(quad 1 1 1 1)

   = quad (2nd (quad 1 1 1 1)) (3rd (quad 1 1 1 1)) (4th (quad 1 1 1 1)) (1st (quad 1 1 1 1))

   = quad 1 1 1 1

6. what is f1 (f1 (quad 2 2 2 2))?

   Answer. f1 (quad 2 2 2 2)

   = (λq. quad (2nd q) (3rd q) (4th q) (1st q))(quad 2 2 2 2)

   = quad (2nd (quad 2 2 2 2)) (3rd (quad 2 2 2 2)) (4th (quad 2 2 2 2)) (1st (quad 2 2 2 2))

   = quad 2 2 2 2

   so, f1 (f1 (quad 2 2 2 2)) = f1 (quad 2 2 2 2) = quad 2 2 2 2

7. what is f1 (f1 (f1 (quad 3 3 3 3)))?

   Answer. f1 (f1 (f1 (quad 3 3 3 3)))

= f1 (f1 (quad 3 3 3 3))

= f1(quad 3 3 3 3)

= quad 3 3 3 3

8. what does the function λn. 1st (n f1 (quad n n n n)) calculate for a Church's numeral n? Give a compact description.
   Answer. The function returns n.

9. what is f2 (tri fls 40 45)?

```
(λt.  tri    ( OR (fst t) (EQUAL (fst(snd t)) (snd (snd t))) )
             ( TIMES (fst (snd t)) 2 )
             (snd (snd t) )
) (tri fls 40 45)

= tri  ( OR (fst (tri fls 40 45)) (EQUAL (fst(snd (tri fls 40 45))) (snd (snd (tri fls 40 45)))) )
       ( TIMES (fst (snd (tri fls 40 45))) 2 )
      (snd (snd (tri fls 40 45)) )
=  tri  ( OR (fls) (EQUAL 40 45) )
        ( TIMES (fst (snd (tri fls 40 45))) 2 )
        (snd (snd (tri fls 40 45)) )
=  tri  ( OR (fls) (fls) )
        ( TIMES (fst (snd (tri fls 40 45))) 2 )
        (snd (snd (tri fls 40 45)) )
= tri   ( fls )
        ( TIMES 40 2 )
        (snd (snd (tri fls 40 45)) )
= tri   ( fls )
        ( 80 )
        ( 45 )
```

What this long calculation does is to compare the second and third elements of the tri. If they are equal the resulting first element will be tru. The third element is not changed and the second element is multiplied by 2.

If we repeat this multiple times, the second element is repeatedly multiplied by 2. If at any point it becomes equal to the third element, the resulting first element becomes tru.

Finally, if the first element is tru and we apply the function, since we are doing an OR, the first element will stay tru.

10. what is f2 (f2 (tri fls 40 45))?

    tri fls 160 45

11. what is f2 (f2 (f2 (tri fls 40 45)))?

    tri fls 320 45

12. what does the function $\lambda n.\ \lambda p.\ \lambda q.$ fst (n f2 (Tri fls p q)) calculate? Give a compact description.

Function returns tru if $p \times 2^i = q$ for some $i \leq n$ and returns fls otherwise

Problem 2 (Linked Lists with Lambda Calculus) In class, we defined a lambda calculus representation of linked lists. I start by repeating the definition and follow that with a number of questions.

A list is represented with pairs. The first part of the pair is an element and the second part of the pair is a list. In order to indicate if the second element represents a non-empty list, we use a Boolean flag. I start by giving a few examples.

empty list pair fls fls

fls indicates that the list is empty, so the second element of the pair should not be accessed

list with one element a = pair tru (pair a (pair fls fls))

tru	indicates that the list is not empty, so the second element of the pair which is (pair a (pair fls fls)) contains the list data
a	is the first element of the list
(pair fls fls)	is the remainder of the list. In this case, is the empty list

list with two elements a and b = pair tru (pair a (pair tru (pair b (pair fls fls))))

tru	indicates that the list is not empty, so the second element of the pair which is (pair a (pair tru (pair b (pair fls fls)))) contains the list data
a	is the first element of the list
(pair tru (pair b (pair fls fls)))	is the remainder of the list. In this case, it is a list that contains one element which is b

In general, the representation of a list with elements $a_1, a_2, a_3, \ldots, a_k$ in the given order, where $\mathbf{k \geq 1}$ is

pair tru (pair a_1 L)

where L is the representation of the list whose elements, in order, are $a_2, a_3, \ldots, a_k$

For all the questions, you are asked to write functions. You should understand that to mean write a lambda expression.

Questions

- Assume that you have a function dropLast that drops the last element of a list. Also, assume that you have a functions firstElem and lastElem that return the first and the last element of a list. Write a function to check if a list of Church's numerals is a palindrome (is the same as its reverse). For Example:
  - The empty list is a palindrome
  - A list of only one element is a palindrome

- The list pair tru (pair 1 (pair tru (pair 2 (pair tru (pair 2 (pair tru (pair 1 (pair fls fls))))))) which represents the list 1, 2, 2, 1 is a palindrome

Answer. A list is a palindrome if it is

1. the empty list
2. A list of only one element
3. A list of two or more elements such that
    a. the first and last element are the same
    b. the middle portion obtained by discarding the first and last elements is also a palindrome

Here is a function that implements the conditions above.

```
g = λplm. λL.  (emptyList n)
                    tru                              // list is empty
                    ( (emptyList (dropFirst L))
                         tru                         // list has only one element

                         // the part below handles a list with more than one
                         // element
                         ( (equal (firstElem n) (lastElem L) )

                              // if the first and last elements are equal
                              // check that the middle part is a palindrome
                              plm (dropLast (dropFirst  L)))

                              // if the first and last elements are not equal
                              // the list is not a palindrome
                              fls
                         )
                    )
palindrome = fix g
```

In the function, (dropFirst L) drops the first element of L. It is only applied to non-empty lists and its equivalent to (snd (snd L))

- Write a recursive function that returns the number of elements that are equal to the first element in a list of Church's numeral. For example

    - The function should return 0 for the empty list
    - The function should return 1 for a list of one element
    - The function should return 4 for the list 1, 2, 1, 3, 1, 5, 1

Note. There was a typo in the original problem (3 instead of 4), but that will not affect the grading as long as your answer has internal consistency.

It would help to first write a function that takes two arguments: an Church's numeral and a list of Church's numeral and returns the number of elements in the list equal to that element. The function will have the form λa. λL. .... and returns the number of elements in L that are equal to a.

Answer.

For a given list with one or more elements, the remainder of the list is the list obtained by dropping the first element.

The rules that the code implements are the following. We have a list and a given element and we want to determine the number of occurrences of the given element I the list.

> If the list is empty the count is 0
> if the list is not empty, the there are two cases
>> the given element is equal to the first element of the list:
>>> in this case, the count is 1 + the number of occurrences of the the given element in the remainder
>>
>> the element we are checking is not equal to the first element of the list
>>> in this case, the count is equal to the number of occurences of the given element in the remainder

Here is the code that implements these ideas

```
g = λcount. λa. λL.  (emptyList L)
                      0
                     ( (equal a  (firsttElem L))
                            ( plus 1 (count a  (dropFirst L)) )
                            (count a (dropFirst L))
                     )
```

Count = fix g

The function we want is therefore

```
NumberOfOccurrencesOfFirstElement = λL. (emptyList L)
                                         0
                                         (count (firstElem L) L)
```

Note that the code can be simplified but because we can apply count even if the list has not elements, but I think this is clearer.

- Write a recursive function to determine if two lists of Church's numerals are equal

Answer.

Two lists are equal if and only if

a. they are both empty or
b. the are both not empty and the their first elements are the same and the lists obtained by dropping their first elements are also equal

Here is the code that implements these ideas

```
g = λeq. λL1. . λL1.  (and  (emptyList L1) (emptyList L2))
                             // if both lists are empty, they are equal
                             tru
                             ( (or   (and  (emptyList L1) (NOT (emptyList L2)) )
                                     (and  (NOT (emptyList L1)) (emptyList L2) ))

                                     // if one list is empty and the other is not empty,
                                     // the two lists are not equal
                                     fls

                                     // if the two lists are not empty, then they must have
                                     // the same first element and the lists obtained by dropping
                                     // the first elements must be equal
                                     (and (equal (FirstElem L1) (FirstElem L2))
                                          (eq (dropFirst L1) (dropFirst L2))
                             )
```

equalLists = fix g

Hint look at the solution of HW3 from Fall 2019, HW4 from Fall 2020 and HW4 from Spring 2021

Problem 3. Pointer Semantics in C. See pdf separate file for solution.