CSE340 FALL 2020 HOMEWORK 4
Due Monday November 16 2020 by 11:59 PM

PLEASE READ THE FOLLOWING CAREFULLY

1. Your answers must be typed.
2. On Gradescope, you should submit the answers to separate question separately.
3. 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

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)
First = λq. fst (fst q)
Second = λq. snd (fst q)
Third = λq. fst (snd q)
Fourth = λq. snd (snd q)
Tri = λx. λy. λz. pair x (pair y z)
next0 = λp. pair **(**PLUS (fst p) (snd p)**) (**PLUS (snd p) (snd p)**)**
next1 = λq. quad (Second q) (Third q) (First q) (Fourth q)
next2 = λt. Tri (OR (fst t) **(**EQUAL **(**fst (snd t)**) (**snd (snd t)**))**) // first element
(TIMES (fst (snd t)) 3) // second element
(TIMES (snd (snd t)) 2) // third element

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

1. What is next0 (pair 1 1)?
2. What is next0 (next0 (pair 1 1))?
3. What is next0 (next0 (next0 (pair 1 1)))?
4. What does the function λn. fst (n next0 (pair 1 1)) calculate? Give a *compact* description.
5. what is next1 (quad fls tru fls tru)?
6. what is next1 (next1 (quad fls tru fls tru))?

7. what does the function λn. First (n next1 (quad fls tru fls tru)) calculate? Give a compact description.
8. what is Tri fls 40 45?
9. what is next2 (Tri fls 40 45)
10. what is next2 (next2 (Tri fls 40 45))
11. what does the function λn. λp. λq. First (n next2 (Tri fls p q)) calculate? Give a compact description.

Problem 2 (Linked Lists with Lambda Calculus) We define a lambda calculus representation of linked lists. 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. We also, use a header in order to be able to represent the empty list. 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 **k ≥ 1** is

pair tru (pair a_1 L)

where L is the representation of the list whose element in order are $a_2, a_3, \ldots, a_k$
Note that only the last pair has "fls" as the first element to indicate the empty list. For all the questions, you are asked to write functions. You should understand that to mean write a lambda expression.

Questions

1. Write a recursive function that shifts a list right. If the input to the function is a list representation of $a_1, a_2, \ldots, a_k$, the output should be a list representation of $a_k, a_1, \ldots, a_{k-1}$
2. write a function that takes as input a list L and a Church numeral n and returns a list which is obtained from L by shifting L to the right n times. You can assume that you have a function that shifts write (part 1) in answering this part.

Hint look at the solution of HW3 from Fall 2019.

Problem 3. Pointer Semantics in C. Consider the following C program

```
#include <stdio.h>
#include <stdlib.h>

struct T {
        int i;
        struct T * next;
};

struct T *b[4];         // Global variable. Locations m1 through m4 are
                        // associated with b[0] through b[3].
struct T **c[4];        // global variable. locations m5 through m8 are
                        // associated with c[0] through c[3].

int main()
{
   b[0] = (struct T *) malloc(sizeof(struct T));     // location m9 allocated
   c[0] = (struct T **) malloc(sizeof(struct T *));  // location m10 allocated
   b[1] = *c[0];

   { struct T *a[4]; // a[0] through a[3] are in locations m11 through m14
     // point 1
     for (int i = 0; i < 3; i++)
     {
           a[i] = (struct T *) malloc(sizeof(struct T));// locations m15 through
                                                        // m17 allocated in
                                                        // successive iterations
           b[i+1] = a[i];
           c[i] = &b[i+1];
           b[i]->next = *c[i];
     }
     // point 2
   }
   // point 3

   free(*c[2]);
   // point 4

   b[2] = b[1];    // assignment 1
   *c[2] = *c[0];  // assignment 2
   //point 5
}
```

Remember

- A wild pointer is a pointer that is not initialized. It is different from a dangling reference in that a dangling reference points to memory that has been previous allocated and then de-allocated.
- When counting wild pointers, we do not count uninitialized fields in previously deallocated locations or in garbage locations.
- When counting dangling references, we do not count fields in previously deallocated locations or in locations that are garbage.
- Global variables are initialized to 0 in C

Questions

1. What are (if any) the dangling references, the wild pointers and the garbage locations at point 1

2. ~~What is an alias of b[1] at point 1? Your answer should be an expression that does not contain "b".~~
3. What are (if any) the dangling references, the wild pointers and the garbage locations at point 2
4. What is an alias of b[0] at point 2? Your answer should be an expression that does not contain "b". This is a little harder than question 2 above.
5. What are (if any) the dangling references, the wild pointers and the garbage locations at point 3
6. What are (if any) the dangling references, the wild pointers and the garbage locations at point 4
7. Executing assignment 1 results in an arrow from which location to which location?
8. Executing assignment 2 results in an arrow from which location to which location?
9. What are (if any) the dangling references, the wild pointers and the garbage locations at point 5