

CSE505 – Spring 2021  
**Assignment 4 Problem 2**  
*(may be done by the same team as for Problem 1)*

*Problem 1 is due on May 3, 11:59 pm.*

**Problem 2: Lambda Calculus (due: May 7, 11:59 pm)**

**(a)** Two important reduction strategies in the lambda-calculus are leftmost-outermost (LO) and leftmost-innermost (LI). For each of the two operations mentioned below, compare LI and LO with respect to the number of reduction steps needed to compute the normal form as a function of the size of the input Church numerals. Assume the Church numeral for number  $n$  is of size  $n$ .

- (i) The 'add' operation – for addition of two Church numerals
- (ii) The 'times' operation – for multiplication of two Church numerals

See file `defs.txt` in the LAMBDA directory posted on Piazza for their definitions.

For the 'add' operation, consider invocations of the form `((add n1) n2)`, where  $n1 \geq 0$  and  $n2 \geq 0$ . For the 'times' operation, consider only invocations of the form `((times n) 1)`, where  $n \geq 0$ .

Use the Lambda Calculus Simulator to run the 'add' and 'times' operations on sample inputs to understand how they work for the two reduction strategies. The simulator accepts Arabic numerals as inputs and converts them to Church numerals internally for performing reductions.

Write a few sentences comparing LI and LO for each of the two operations. Be sure to state the number of steps each strategy takes for each of the two operations as a function of the size of the input Church numerals. Write your answer in a file `compare.txt`.

**(b)** Develop in the lambda-calculus a representation for binary trees following the technique outlined in Lecture 25, slides 7-10. Show the representation for the following binary tree:

```
node(node(node(leaf(NY), leaf(PA)), leaf(MA)), node(leaf(OH), leaf(CT)))
```

Write your answer in a file `tree.txt` by introducing a 'let' definition of the form:

```
let tree = _____
```

Next, define in the lambda-calculus a function that counts the number of leaf nodes in a binary tree. Write your answer in file `tree.txt` by adding a 'let' definition of the form:

```
let leafcount = Lt. _____
```

**Hint:** (1) You do *not* need to start with a recursive definition. A concise non-recursive definition is possible similar to – but not exactly the same as – the 'size' function for lists.

(2) You may include, as a 'helper' function, one of the arithmetic operations from `defs.txt`.

Test your answer by deriving the normal form of  $(\text{leafcount } \text{tree})$ . Your answer should be the Church numeral for 5.

#### **WHAT TO SUBMIT:**

Prepare a top-level directory named *A4\_Problem2\_UBITId1\_UBITId2* if the assignment is done by two students; otherwise, name it as *A4\_Problem2\_UBITId* if the assignment is done solo. (Order the *UBITId*'s in alphabetic order, in the former case.) In this directory, place the files `compare.txt` and `tree.txt`.

Compress the directory and submit the compressed file using the online submission procedure – instructions posted at Resources → Assignments → Online\_Submission.pdf. Only one submission per team is required.

**End of Assignment 4 Problem 2**