

Problem Set 2
COMP301 Fall 2020
Week 4: 26.10.2020 - 30.10.2020

Please use the code boilerplate, which includes several tests for you to see if your code is correct. Save your code in the racket file, and draw the abstract syntax tree to a paper/online environment and take the picture/screenshot of it. Zip these two files and name the ZIP file as ID_username.zip with your ID and username (Example: 1234567_etezcan19.zip), and submit this ZIP file. You are expected to submit by the end of PS, however, you have an additional 1 hour to submit after the PS. The solutions will be available on the course BlackBoard after Friday. **Read the questions carefully. Good luck!**

Problem 1: Draw the abstract syntax trees of the following let and lambda-calculus expressions:

a)

```
let x = 3 in let x = -(x,1) in x
```

b)

```
(lambda (x)
  (lambda (y)
    ((lambda (x) (x y))
     x)))
```

Problem 2: Write a procedure called `path` that takes an integer `n` and a binary search tree (BST)¹, which returns the path to `n`. You may assume `n` only exists once. In the code we provide, you can see the example inputs and outputs, so you will not bother creating a BST. The output should be:

- ' () if `n` is not in the BST
- ' (root) if `n` is the root
- the path to `n` if `n` is deeper in the BST (*see examples in the code we provide*)

Hint: Appending an empty list to some non-empty list results in the same non-empty list.

Problem 3: Explain what the LET expressions below evaluate to. You can also say the expression is wrong and gives an error, if you think that is the case.

a)

```
let x = let y = 3 in x in 3
```

b)

```
let x = 3 in let y = -(-(0,x),1) in -(y,x)
```

c)

```
if zero?(0) then if zero?(1) then 1 else 0 else zero?(0)
```

Food for thought: How does the language know to which if the first else belongs to?. Would this be a problem if using else was optional?

¹EOPL p.10