

Problem Set 5
Comp 301
Fall 2022
Week 6: 07.11.2022 - 11.11.2022

Instructions:

- Submit your answers to the Blackboard PS5 assignment until November 11st Saturday, at 23.59.
- Please use the code boilerplate for LET language, which includes several tests for you to see if your code is correct.
- Save your code, write your answers for the first two questions on a piece of paper or online environment and include it as a PDF. Zip your code along with the PDF as ID_username.zip with your ID and username (Example: 1234567_otal19.zip), and submit this ZIP file.
- Read the questions carefully. Good luck!

Problem 1: ¹: Write out the derivation of the following program as a derivation tree. Check figures 1 and 2 as examples.

Let $\rho = [x = [20], y = [15], z = [10]]$

if **zero?** $(-(z, y))$ then $-(x, z)$ else $-(12, x)$

Example 1: **if** **zero?** (0) then 5 else 2

$$\frac{(\text{value-of } \langle\text{zero?}(0)\rangle \rho = (\text{bool-val } t))}{(\text{value-of } \langle\text{if zero?}(0) \text{ then } 5 \text{ else } 2 \rangle \rho) = 5}$$

FIGURE 1. Derivation tree for "if zero? $(-(y, 10))$ then 5 else 2"

Example 2: **zero?** $(-(x, y))$

$$\frac{\frac{(\text{value-of } \langle x \rangle \rho) = 20 \quad (\text{value-of } \langle y \rangle \rho) = 15}{(\text{value-of } \langle -(x, y) \rangle \rho) = 5}}{(\text{value-of } \langle \text{zero?}(-(x, y)) \rangle \rho) = (\text{bool-val } \#f)}$$

FIGURE 2. Derivation tree for "zero? $(-(x, y))$ "

Problem 2: Now, we want you to fill in the blanks for the output of this interpreter. Similar to first question let $\rho_0 = [x = [20], y = [15], z = [10]]$. Note that below snippet is not complete, although we only want answers to the _____'s, we recommend you to continue the evaluation as a self study exercise.

¹Similar to EOPL p.70 Exercise 3.4

```

let n = 15 in if zero?(-(x, n)) then -(x, y) else -(-(z, n), -(y, x))
(value-of
  <<let n = 15
    in if zero?(-(x, n)) then
      -(x, y)
    else -(-(z, n), -(y, x))>>
  ρ0)

= (value-of
  <<if zero?(-(x, n)) then
    -(x, y)
    else -(-(z, n), -(y, x))>>
  ____[1]____)

Let ρ1 = ____[2]____

= (if (expval->bool (value-of <<zero?(-(x, n))>> ρ1))
  (value-of <<-(x, y)>> ρ1)
  (value-of <<-(z, n), -(y, x)>> ρ1))

= (if ____[3]____
  (value-of <<-(x, y)>> ρ1)
  (value-of <<-(z, n), -(y, x)>> ρ1))

= (value-of <<____[4]____>> ρ1)

= [(-
  [(value-of <<____[5]____>> ρ1) ]
  [(value-of <<____[6]____>> ρ1) ])]

= ____[7]____

```

Problem 3:² Extend the Let language by adding the following new operations minus, double, and gcd and prime?. You need to work with Let language codes for this problem.

Hint: You only need to change lang.rkt and interp.rkt.

Hint: For part C and D you can write a helper functions for calculating the gcd of two numbers and prime number checker.

Part A. Extend the Let language by adding a new operation minus that takes one argument n and returns -n. For example, the value of minus(-(minus(5), 9)) should be 14, which is equal to -((-5) - 9).

Part B. Extend the Let language by adding the double operation, which takes a single number as an input and returns the same number multiplied by 2. For example double(3) must return 6.

Part C. In this part, you will need to add the gcd operation to the Let language. gcd is short for Greatest Common Divisor. The gcd of two integers x and y is the greatest number which is a divisor for both x and y. For example, gcd(12, 16) is equal to 4. For more about the gcd you can check this link

²EOPL p.72 Exercise 3.6

Part D. In this part, you will add the `prime?` operation to the `Let` language. The `prime?` will check if the given input integer `x` can be divisible by a smaller number than input bigger than zero. For example, `prime? (2)` will return output `#t`, whereas `prime? (4)` returns output `#f`

After you finish implementing the described operations, uncomment and run the tests in `tests.rkt`. At the end of that file we defined extra tests for `minus`, `double`, and `gcd` you can use this as an example to how to write test cases and define your own tests.