Nazir Nayal 62775
Moayed Haji Ali 66688

Project 2

COMP 301
November 15, 2020

Part A

(1)

- Syntax Data Types
- Values
- Environment
- Behaviour Specification
- Behaviour Implementation

(2)

- Syntax : **lang.rkt**
- Data Types: **data-structures.rkt**
- Values: **data-structures.rkt**
- Environment: **environments.rkt**
- Behaviour Specification: **interp.rkt**
- Behaviour Implementation: **interp.rkt**

Part B

(1)

we define an initial environment that contains the variables x = 1, y = 5, z = 10 by extending an empty environment three times:

```
(define init-env
  (lambda ()
    (extend-env
     'x (num-val 1)
     (extend-env
      'y (num-val 5)
      (extend-env
       'z (num-val 10)
       (empty-env))))))
```

(2)

$\rho = []$
$\rho_1 = [\text{z} = 10]\rho \rightarrow [z = 10]$
$\rho_2 = [\text{y} = 5]\rho_1 \rightarrow [z = 10, y = 5]$
$\rho_3 = [\text{x} = 1]\rho_2 \rightarrow [z = 10, y = 5, x = 1]$

Part C

- Expressed Values: Bool + Int + String
- Denoted Values: Bool + Int + String

Part D

(4)

Custom Expression Explanation: The expression we have implemented is called *reverse*. It receives an expression that evaluates to a string, and returns a reversed version of this string.

```
Expression ::= reverse expression
[rev-exp (str)]
```

Team workload breakdown:

Moayed Haji Ali:

Defined and implemented the grammar, str-exp, op-exp, and solved part B. In the bonus project, defined the grammar, the binary tree data-type, Ropes, concatenation, and fetching i_{th} character.

Nazir Nayal:

Implemented if-exp, custom expression and solve part A and C. In the bonus project, implemented sub-string and re-balancing procedures.