

COMP 301 Project 2

Ahmet Uyar, 72847
Mehmet Rüçhan Ortayatırtmacı

Workload Breakdown

Ahmet Uyar - Part D
Mehmet Rüçhan Ortayatırtmacı - Part A, B, C

Part A)

(1) Write the 5 components of the language:

- a) Syntax and datatypes
- b) Values
- c) Environment
- d) Behavior Specification
- e) Behavior Implementation

(2) For each component, specify where or which racket file (if it applies) we define and handle them.

- a) **Syntax:** lang.rkt, **Datatypes:** data-structures.rkt
- b) **Values:** data-structures.rkt
- c) **Environment:** definition: data-structures.rkt , related procedures: environments.rkt
- d) **Behavior Specification** - we specify the behavior of each expression by:
 - (i) constructors
 - (ii) observer, namely the behavior of value-of

In our language, a program is essentially an expression. To determine its value, we must define the values of any free variables within the program, known as the initial environment. Therefore, the value of a program is determined by evaluating the expression within the initial environment.

- e) **Behavior Implementation:** we implement the behavior specification at interp.rkt with value-of.

Part B)

(1) Create an initial environment that contains 3 different variables (x, y, and z).

It is defined in **environments.rkt** and x, y, z initialized as 4, 2, 3.

(init-env) = [x=4, y=2, z=3]

```
(define init-env
  (lambda ()
    (extend-env 'x (num-val 4)
      (extend-env 'y (num-val 2)
        (extend-env 'z (num-val 3) (empty-env))))))
```

(2) Using the environment abbreviation shown in the lectures, write how the environment changes at each variable addition.

[] **p0** : after empty-env

[z=3]**p1**: after (extend-env 'z (num-val 3) (empty-env)))))

[y=2, z=3]**p2**: after (extend-env 'y (num-val 2) (extend-env 'z (num-val 3) (empty-env)))))

[x=4, y=2, z=3]**p3**: after (extend-env 'x (num-val 4) (extend-env 'y (num-val 2) (extend-env 'z (num-val 3) (empty-env)))))

(3) There are different representations that we use to implement our environments. What are those implementations and which one we used in MYLET language?

- 1) Data Structure Representation
- 2) Procedural Representation

MYLET uses the Data Structure Representation

Part C)

What is the difference between expressed and denoted values? Specify expressed and denoted values for MYLET language.

Expressed values are the actual results of expression evaluations, while denoted values are the abstract meanings assigned to expressions within the language's semantics such as whether the data is a list, a boolean, a string, or a number.

$$\begin{aligned} \text{ExpVal} &= \text{Int} + \text{Bool} + \text{List}<\text{Int}> + \text{Pair}<\text{Int}, \text{Int}> \\ \text{DenVal} &= \text{ExpVal} \end{aligned}$$

Part D)

All test cases pass without any errors