

Man Tik Li
CS 360 – 002
Krzysztof Nowak
July 14, 2020

Lab 2

Serial Number: C02NLN90G3QJ

OS: macOS Catalina

Hardware UUID: 793DBFBA-8B38-5104-9E39-7BBF399D3196

Tested and implemented in DrRacket (version 7.7) and Tux.

1.

- i. The function `force` takes `L` as a promise, the promise is forced to obtain a value. If the promise has not been forced before, then the result is recorded in the promise so that future forces on the promise produce the same value. When run the `(take 5 L)`, the `L` is forced to obtain 1-5 value and recorded it in the promise. If run `(take 7 L)` after `(take 5 L)` because 1-5 value already recorded. Therefore, it only prints 6 and 7.

```
[ml3546@tux4:~/CS360/lab_2_code$ scheme
MIT/GNU Scheme running under GNU/Linux
Type '^C' (control-C) followed by 'H' to obtain information about interrupts.

Copyright (C) 2019 Massachusetts Institute of Technology
This is free software; see the source for copying conditions. There is NO warranty; not even for
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Image saved on Thursday September 5, 2019 at 11:51:46 AM
Release 10.1.10 || Microcode 15.3 || Runtime 15.7 || SF 4.41 || LIAR/x86-64 4.118

[1 ]=> (load "delayedmap.scm")

;Loading "delayedmap.scm"... done
;Value: 1

[1 ]=> (take 5 L)
1
2
3
4
5
;Value: (1 2 3 4 5)

[1 ]=> (take 7 L)
6
7
;Value: (1 2 3 4 5 6 7)
```

- ii. To verify that the height of the tree is two, there should be four leaves in the tree.

```
ml3546@tux3:~/CS360/lab_2_code$ scheme
MIT/GNU Scheme running under GNU/Linux
Type ^C (control-C) followed by ^H to obtain information about interrupts.

Copyright (C) 2019 Massachusetts Institute of Technology
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Image saved on Thursday September 5, 2019 at 11:51:46 AM
Release 10.1.10 || Microcode 15.3 || Runtime 15.7 || SF 4.41 || LIAR/x86-64 4.118

1 ]=> (load "BST.scm")

;Loading "BST.scm"... done
;Value: (3 (1 () (2 () ())) (6 () ()))

1 ]=> (define new_BST '())
(define new_BST (insert 4 new_BST))
(define new_BST (insert 6 new_BST))
(define new_BST (insert 5 new_BST))
(define new_BST (insert 7 new_BST))
(define new_BST (insert 2 new_BST))
(define new_BST (insert 1 new_BST))
(define new_BST (insert 3 new_BST))
;Value: new_bst

1 ]=>
;Value: new_bst

1 ]=>
;Value: new_bst

1 ]=>
;Value: new_bst

1 ]=>
;Value: new_bst

1 ]=>
;Value: new_bst

1 ]=>
;Value: new_bst

1 ]=> (begin new_BST)
;Value: new_bst

1 ]=>
;Value: (4 (2 (1 () (3 () ())) (6 (5 () (7 () ())))))

1 ]=> █
```

2. Eva and Louis have a different system version of map. The system versions of car, cdr, and apply are not the correct versions. For instance, the metacircular evaluator will not work on procedures if the apply is different from the system.
3. Incompleted

lab2.rkt - DrRacket

Check Syntax Debug Macro Stepper Run Stop

```
1 #lang racket
2 (require rackunit)
3 (require rackunit/text-ui)
4
5 ; Function square to preforms square calculation
6 (define (square x) (* x x))
7
8 ; Function fib compute the Fibonacci numbers
9 (define (fib n)
10   (fib-iter 1 0 0 1 n))
11
12 ; Function fib-iter recall the transformation to the state variables a and b
13 (define (fib-iter a b p q count)
14   (cond ((= count 0) b)
15         ((even? count)
16          (fib-iter a
17                    b
18                    (+ (square p) (square q)) ; compute p'
19                    (+ (* 2 p q) (square q)) ; compute q'
20                    (/ count 2)))
21         (else
22          (fib-iter (+ (* b q) (* a q) (* a p))
23                    (+ (* b p) (* a q))
24                    p
25                    q
26                    (- count 1)))))
27
28 ; Test
29 (define lab2
30   (test-suite
31     "Lab2"
32     (test-case "fib"
33       (check-equal? (fib 0)
34                     0)
35       (check-equal? (fib 1)
36                     1)
37       (check-equal? (fib 9)
38                     34)
39       (check-equal? (fib 10)
40                     55)
41       (check-equal? (fib 100)
42                     354224848179261915075)
43     )))
44
```

Welcome to DrRacket, version 7.7 [3m].
Language: racket, with debugging; memory limit: 512 MB.
1 success(es) 0 failure(s) 0 error(s) 1 test(s) run

Determine language from source

27.0 592.27 MB

4.