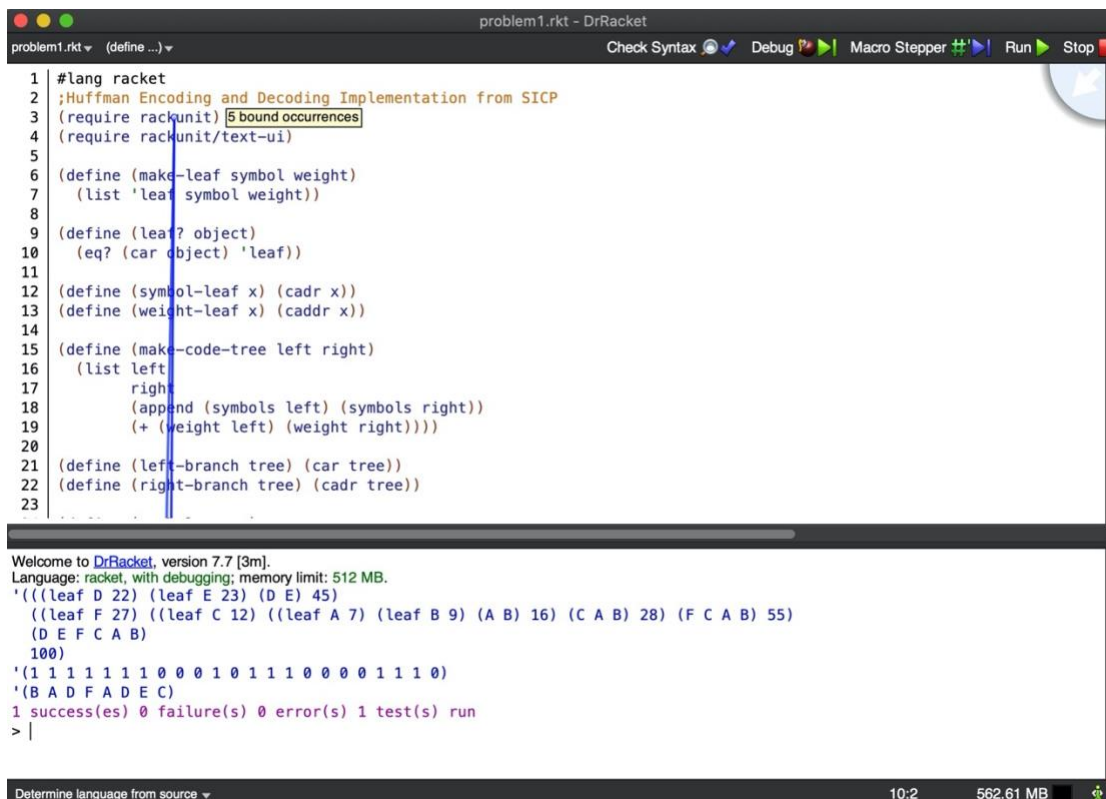


Man Tik Li
CS 360
Krzysztof Nowak
August 3, 2020

Assignment 2

Tested and implemented in Racket v.7.7

The program was implemented from source code within SICP as well as the SICP scheme code provided as a reference. The creation of a Huffman tree works as expected. The nodes and leaves are displayed. The encoding and decoding of the message work as expected by displaying the original message. A test case that checks for the correctness.



```
1 #lang racket
2 ;Huffman Encoding and Decoding Implementation from SICP
3 (require rackunit)
4 (require rackunit/text-ui)
5
6 (define (make-leaf symbol weight)
7   (list 'leaf symbol weight))
8
9 (define (leaf? object)
10  (eq? (car object) 'leaf))
11
12 (define (symbol-leaf x) (cadr x))
13 (define (weight-leaf x) (caddr x))
14
15 (define (make-code-tree left right)
16   (list left
17         right
18         (append (symbols left) (symbols right))
19         (+ (weight left) (weight right))))
20
21 (define (left-branch tree) (car tree))
22 (define (right-branch tree) (cadr tree))
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Welcome to DrRacket, version 7.7 [3m].
Language: racket, with debugging; memory limit: 512 MB.

```
'(((leaf D 22) (leaf E 23) (D E) 45)
 ((leaf F 27) ((leaf C 12) ((leaf A 7) (leaf B 9) (A B) 16) (C A B) 28) (F C A B) 55)
 (D E F C A B)
 100)
'(1 1 1 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 0 1 1 1 0)
'(B A D F A D E C)
1 success(es) 0 failure(s) 0 error(s) 1 test(s) run
> |
```

Determine language from source 10:2 562.61 MB