

# CS2200 Homework 3

Evan Wilcox

Due February 26, 2019

1. Fully recursive merge sort implemented in Python 2.7.

```
8  # 1a
9  def Merge(L, R):
10     if len(L) == 0:
11         return R
12     elif len(R) == 0:
13         return L
14     elif L[0] < R[0]:
15         return L[:1] + Merge(L[1:], R)
16     else:
17         return R[:1] + Merge(L, R[1:])
18
19 # 1b
20 def MergeSort(A):
21     if len(A) <= 1:
22         return A
23     else:
24         m = len(A)/2
25         return Merge(MergeSort(A[:m]), MergeSort(A[m:]))
26
```

2. Define an operation  $\div$  called *bounded subtraction* as follows:

- (a)  $0 \div a = 0$
- (b)  $a \div 0 = a$
- (c)  $a' \div b' = a \div b$

Prove that for all  $a, b$  in  $\mathbb{N}$ ,  $(a + b) \div b = a$ .

Let  $Q = \{q \in \mathbb{N} \mid (a + q) \div q = a\}$

$Q$  is not empty,  $0 \in Q$  because  $(a + 0) \div 0 = a$

Start	Current	Reason
$(a + q') \div q'$	$= (a + q')' \div q'$	Definition of $+$
	$= (a + q) \div q$	Definition of $\div$
	$= a$	$q \in Q$

3. Simulator built in Python 2.7 that was used to test the .fsa files for problems 4 and 5.

A lines contain an A followed by a space followed by a string containing all the characters in the alphabet. There can only be one A line.

S lines contain an S followed by a space followed by the name of a state followed by a comma and a space followed by a 0 or 1 if the state is a final state.

B lines contain a B followed by a space followed by the name of the starting state. There can only be one B line.

D lines contain a D followed by a space followed by a state followed by a comma and a space followed by a character in the alphabet followed by a comma and a space followed by the next state.

T lines contain a T followed by a space followed by the tape to be tested.

O lines contain an O followed by a space. O lines must come directly after T lines.

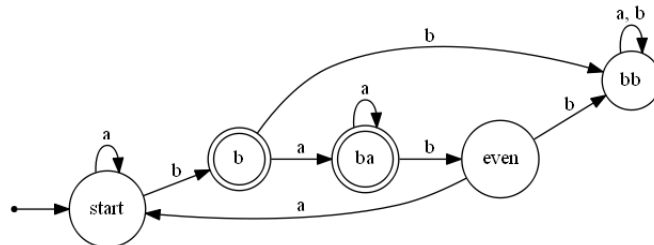
Lines must come in the order described above.

```

6 class Simulator():
7
8     def __init__(self, file):
9         f = open(file, "r")
10
11         self.states = []
12         self.delta = []
13         outputs = []
14         l = []
15
16         for line in f:
17             l.append(line)
18
19             if line[0] == 'A':
20                 self.alphabet = line[2:]
21
22             if line[0] == 'S':
23                 self.states.append([line[2:len(line)-4], int(line[len(line)-2:len(line)-1])])
24
25             if line[0] == 'B':
26                 self.beginState = line[2:-1]
27
28             if line[0] == 'D':
29                 s = line[2:line.index(',')]
30                 c = line[line.index(',')+2:line.index(',', line.index(',')+1)]
31                 e = line[line.index(',', line.index(',')+1)+2:-1]
32                 self.delta.append([s, c, e])
33
34             if line[0] == 'T':
35                 t = line[2:-1]
36                 o = self.run(t)
37                 outputs.append(o)
38
39         f.close()
40
41         w = open(file, 'w')
42         for line in l:
43             if line[0] == 'O':
44                 line = line[:-1] + outputs[0] + line[-1:]
45                 outputs = outputs[1:]
46                 w.write(line)
47             else:
48                 w.write(line)
49
50         w.close()
51
52     def run(self, tape):
53         state = self.beginState
54         for c in tape:
55             for d in self.delta:
56                 if d[0] == state and d[1] == c:
57                     state = d[2]
58                     break
59
60         if [state, 1] in self.states:
61             return "Accepted"
62         else:
63             return "Rejected"
64

```

4. Find a deterministic finite-state automaton that recognizes the language,  $L$ , consisting of all strings in  $\{a, b\}^*$  that contain an odd number of  $b$ 's such that there is at least one "a" between every two  $b$ 's in the string.

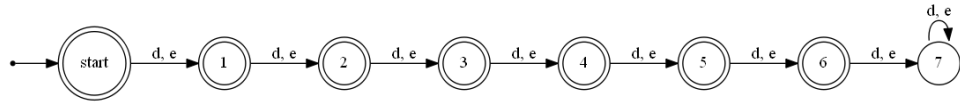


```

A ab
S start, 0
S b, 1
S ba, 1
S bb, 0
S even, 0
B start
D start, a, start
D start, b, b
D b, a, ba
D b, b, bb
D bb, a, bb
D bb, b, bb
D ba, a, ba
D ba, b, even
D even, a, start
D even, b, bb
T a
O Rejected
T b
O Accepted
T ab
O Accepted
T ba
O Accepted
T aba
O Accepted
T bab
O Rejected
T abaaaaba
O Rejected
T baaaaababa
O Accepted
T bb
O Rejected
T abbabbaba
O Rejected

```

5. Let  $L$  be a language over the alphabet  $\{d, e\}$  be the language of all strings having length  $\leq 6$ . Construct a deterministic finite-state machine that recognizes  $L$ .



```

A de
S start, 1
S 1, 1
S 2, 1
S 3, 1
S 4, 1
S 5, 1
S 6, 1
S 7, 0
B start
D start, d, 1
D start, e, 1
D 1, d, 2
D 1, e, 2
D 2, d, 3
D 2, e, 3
D 3, d, 4
D 3, e, 4
D 4, d, 5
D 4, e, 5
D 5, d, 6
D 5, e, 6
D 6, d, 7
D 6, e, 7
T
O Accepted
T d
O Accepted
T e
O Accepted
T dd
O Accepted
T ee
O Accepted
T dedede
O Accepted
T ededed
O Accepted
  
```

T eeeee  
0 Accepted  
T dddddd  
0 Accepted  
T eeeeedddd  
0 Rejected  
T ddededdeed  
0 Rejected