COMP 150 NLP

Fall 2016

Problem Set 1

Mingzhe Li

1.1

The first digit must be either 1 or optional zero. When the first is 1, the second must be 0-2.

/1[0-2]|0?[1-9]/

1.2

The first digit must be 1-3 or optional zero. When the first is 3, the second must be 0-1.

/3[0-1]|[1-2][0-9]|0?[1-9]/

1.3

The year will be either 19xx or 20xx

/(19|20)\d{2}/

1.4

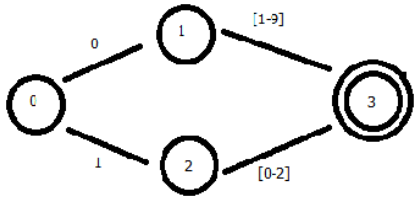
/[- /]/

We can simply combine the above and add separators to get following:

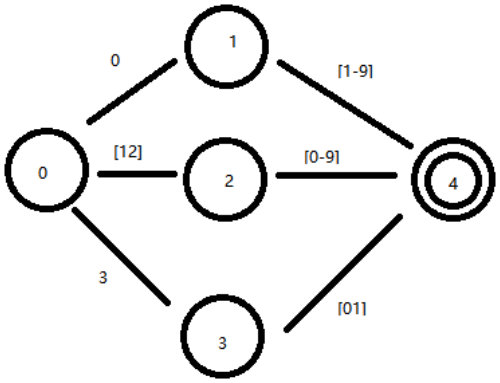
/(0?[1-9]|1[012])[- /](0[1-9]|[12][0-9]|3[01])[- /](19|20)\d{2}/

2. FSA

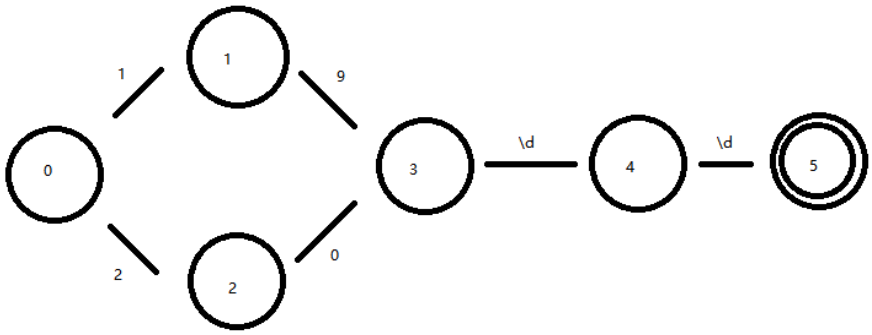
MM:



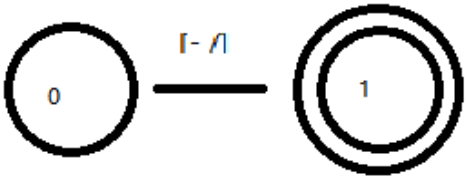
DD:



YYYY:



Separators:



3.

I pretty much followed the algorithm, and thanks to the pseudo code.

My code is in line with my FSA design in Q2.

After long, long hours staring at the screen, I got the final test results:

```
Test Months FSA
''    False
'0'   False
'1'   True
'9'   True
'10'  True
'11'  True
'12'  True
'13'  False

Test Days FSA
''    False
'0'   False
'1'   True
'9'   True
'10'  True
'11'  True
'21'  True
'31'  True
'32'  False

Test Years FSA
''    False
'1899' False
'1900' True
'1901' True
```

```
'1999' True
'2000' True
'2001' True
'2099' True
'2100' False


Test Separators FSA
''    False
','   False
' '   True
'-'   True
'/'   True
'//'  False
':'   False
Process finished with exit code 0
```

4.

The concatenation part: the function copied the transitions from fsa1 to fsa, and then increased the node value by the number of nodes in fsa1 and added the transitions from fsa2 into fsa one by one

I spent multiple hours fighting with Python over a small, simple bug. Python makes type define so flexible that debugging is even more difficult.

Test results:

```
Test Date Expressions FSA
''    False
'12 31 2000' True
'12/31/2000' True
'12-31-2000' True
```

```
'12:31:2000' False
'1 2 2000'   True
'1/2/2000'   True
'1-2-2000'   True
'1:2:2000'   False
'00-31-2000' False
'12-00-2000' False
'12-31-0000' False
'12-32-1987' False
'13-31-1987' False
'12-31-2150' False

Process finished with exit code 0
```

I also created my own test cases, and it passed all tests. Everything should have been implemented correctly. I enjoyed this assignment. Thanks!