0.1 Money FSA

Money FSA

0.2 Regular expressions

In [1]: import re

```python
def do_searches (res, examples):
    for i,re_pat in enumerate(res,1):
        banner = 're%d %s' % (i, re_pat)
        print
        print '%-50s %s' % (banner,'Match')
        print '%-55s %s' % (' '*5 + '='*(len(banner)+1), '='*5)
        print
        for (i,ex) in enumerate(examples,14):
            match = re.search(re_pat,ex)
            if match:
                print ' %2d. %-50s %s' % (i,ex,match.start()->match.end())
            else:
                print ' %2d. %-50s %s' % (i,ex,None)
```

In [9]: import re

# Some regular expressions
#exercise_one_
re1 = r'\[a-zA-Z]+\'

#exercise_two
re2 = r'\[a-z]*d\'

#exercise three. Need word boundaries. See below.
# \b should not be in capture parens.
re3 = r'\b(\w+)\b\s+\b1\b'
re4 = r'(?:\bcharge\b.+\bcall\b)|(?:\bcall\b.+\bcharge\b)'
res0 = [re1, re2, re3, re4]

### Some example strings ###
example1 = 'abracadabra'
example2 = '1billygoat'
example3 = 'billygoat1'
example4 = 'bad apples'
example5 = 'some really really bad apples'

# Does not contain two consecutive repeated words
example6 = 'The doctor tore his sleeve.'
example7 = 'road_runner'
example8 = 'Those doctors charge a fortune for a house call.'
example9 = 'that doctor charged a fortune for that house call.'
example10 = 'The trumpet call signaled the soldiers to charge.'
example11 = 'sad!'
example12 = '*&%#!?'
example13 = 'John believes that that approach will fail.'

examples0 = [example1, example2, example3, example4, example5, example6, example7, example8, example9, example10, example11, example12, example13]

do_searches (res0, examples0)

re1 [a-zA-Z]+ Match
============= =======
14. abracadabra abracadabra
15. 1billygoat  
16. billygoat1  
17. bad apples  
18. some really really bad apples  
19. The doctore tore his sleeve.  
20. road_runner  
21. Those doctors charge a fortune for a house call.  
22. that doctor charged a fortune for that house call.  
23. The trumpet call signaled the soldiers to charge.  
24. sad!  
25. *%#!?  
26. John believes that that approach will fail.  

```
re2 [a-z]*d  
Match

re3 \b(\w+)\b\s+\b\1\b  
Match

re4 (?:\bcharge\b.+\bcall\b)|(?:\bcall\b.+\bcharge\b)  
Match
```
14. abracadabra
15. ibillygoat
16. billygoat1
17. bad apples
18. some really really bad apples
19. The doctore tore his sleeve.
20. road_runner
21. Those doctors charge a fortune for a house call.
22. that doctor charged a fortune for that house call.
23. The trumpet call signaled the soldiers to charge.
24. sad!
25. *&%#!?
26. John believes that that approach will fail.

In [20]: import re

#exercise four
re1 = r'^1+(01+)*$'
#exercise four alternative
re2 = r'1+(01+)*'
#exercise four wrong
re3 = r'^1*(101+)*$'
res1 = [re1, re2, re3]

#exercise four
#exercise four alternative
#exercise four wrong

# Some example strings
example14 = '1001'
example15 = '101101'
example16 = '11010101'
example17 = '101010'
example18 = '10101'
examples1 = [example14, example15, example16, example17, example18]

do_searches (res1, examples1)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>1001</td>
<td>None</td>
</tr>
<tr>
<td>15</td>
<td>101101</td>
<td>101101</td>
</tr>
<tr>
<td>16</td>
<td>11010101</td>
<td>11010101</td>
</tr>
<tr>
<td>17</td>
<td>101010</td>
<td>None</td>
</tr>
<tr>
<td>18</td>
<td>10101</td>
<td>10101</td>
</tr>
</tbody>
</table>

### re2 \(1^+(01^+)^*\)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>1001</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>101101</td>
<td>101101</td>
</tr>
<tr>
<td>16</td>
<td>11010101</td>
<td>11010101</td>
</tr>
<tr>
<td>17</td>
<td>101010</td>
<td>10101</td>
</tr>
<tr>
<td>18</td>
<td>10101</td>
<td>10101</td>
</tr>
</tbody>
</table>

### re3 \(\sim 1^*(101^+)^\$\)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>1001</td>
<td>None</td>
</tr>
<tr>
<td>15</td>
<td>101101</td>
<td>101101</td>
</tr>
<tr>
<td>16</td>
<td>11010101</td>
<td>None</td>
</tr>
<tr>
<td>17</td>
<td>101010</td>
<td>None</td>
</tr>
<tr>
<td>18</td>
<td>10101</td>
<td>None</td>
</tr>
</tbody>
</table>

### 0.3 Mystery FSA

![Mystery FSA Diagram](image)

Mystery FSA

(aba?)+

### 0.4 Minimum Edit Distance

#### 0.4.1 leda vs deal

```
# leda
1 4
```
D(2,1) = 3, D(3,1) = ?

D(3,2) = \min(D(3,1) + 1, D(2,1) + 2, D(2,2) + 1)

D(2,2) = \min(D(2,1) + 1, D(1,1) + 0, D(1,2) + 1)

Total: 4
0.4.2 Drive vs brief

```
# drive
f 5 6 5 4 5 4
e 4 5 4 3 4 3
i 3 4 3 2 3 4
r 2 3 2 3 4 5
b 1 2 3 4 5 6
# 0 1 2 3 4 5
# drive
```

d r i v e 0
b r i 0 e f
---
2 0 0 1 0 1

Total: 4

0.4.3 Drive vs divers

```
# drive
s 6 5 4 5 4 3
r 5 4 3 4 3 2
e 4 3 4 3 2 1
v 3 2 3 2 1 2
i 2 1 2 1 2 3
d 1 0 1 2 3 4
# 0 1 2 3 4 5
# drive
```

d r i v e 0 0
d 0 i v e r s
---
0 1 0 0 0 1 1

Total: 3

In []: