Homework 1

Shruti Agrawal(sxa178830)

1. Regular Expressions (50 points)

Note: Used python re library for executing the regular expression for question 1.1 to 1.6

1.1. the set of all alphabetic strings:

```
line = ["why that is gr8!", "No, it is not gr8 at all!"]
for str in line:
  print(re.findall("[a-zA-Z\s]+", str))
#['why that is gr']
#['No', ' it is not gr', ' at all']
```

1.2. the set of all alphabetic words:

```
line = ["why that is gr8!", "No, it is not gr8 at all!"]
for str in line:
  print(re.findall(r"\b[a-zA-Z]+\b", str))
#['why', 'that', 'is']
#['No', 'it', 'is', 'not', 'at', 'all']
```

1.3. the set of all lower case alphabetic strings ending in a b:

```
line = ["Many programming languages provide regex capabilities, built-in, or via libraries.","Please use tab."]
for str in line:
  print(re.findall(r"[a-z]*b", str))
#['capab', 'b', 'lib']
#['tab']
```

1.4. the set of all lower case alphabetic words ending in a b:

```
line = ["Many programming languages provide regex capabilities, built-in, or via libraries.","Please use tab."]
for str in line:
  print(re.findall(r"\b[a-z]*b\b", str))
#[]
#['tab']
```

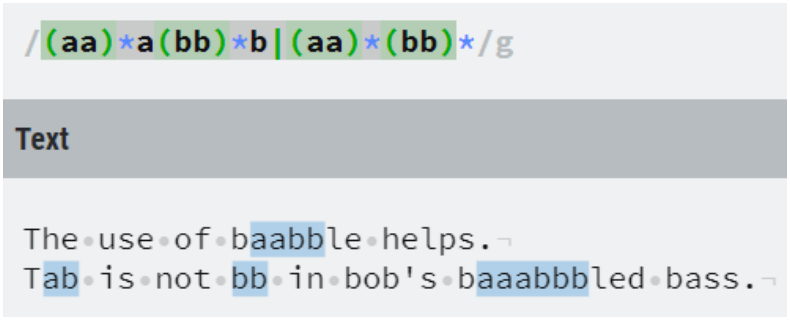
1.5. the set of all strings from the alphabet {“a”, “b”} such that each “a” is immediately preceded by and immediately followed by at least one “b”:

```
line = ["The use of babble helps.","Tab is not bob's bbabled bass."]
for str in line:
  print(re.findall(r"b+ab+", str))
#['babb']
#['bbab']
```

1.6. the set of all words from the alphabet {“a”, “b”} such that each “a” is immediately preceded by and immediately followed by at least one “b”

```
line = ["The use of babb helps.","Tab is not bob's bbabled bab."]
for str in line:
  print(re.findall(r"\bb+ab+\b", str))
#['babb']
#['bab']
```

1.7. the set of all strings from the alphabet {“a”, “b”} that form the pattern an b m where (n+m) is even; n>=0 and m>=0

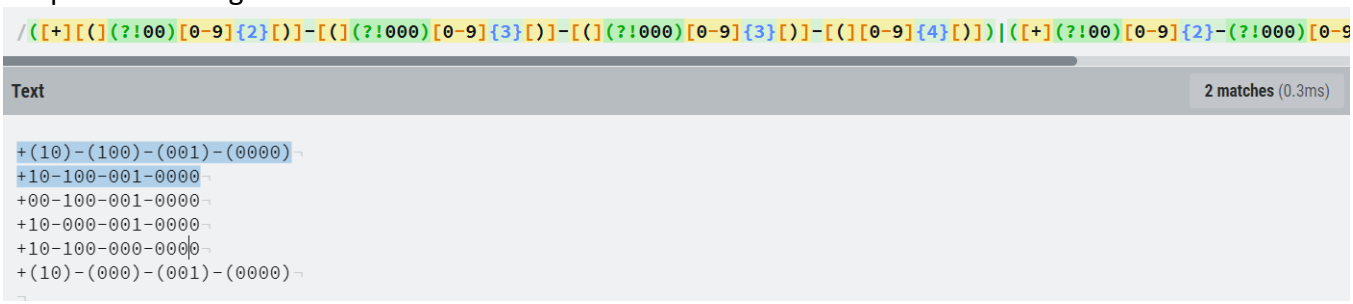


Check for even numbers of a followed by even numbers of b OR check for odd numbers of a followed by odd numbers of b. (Executed on regexr.com)

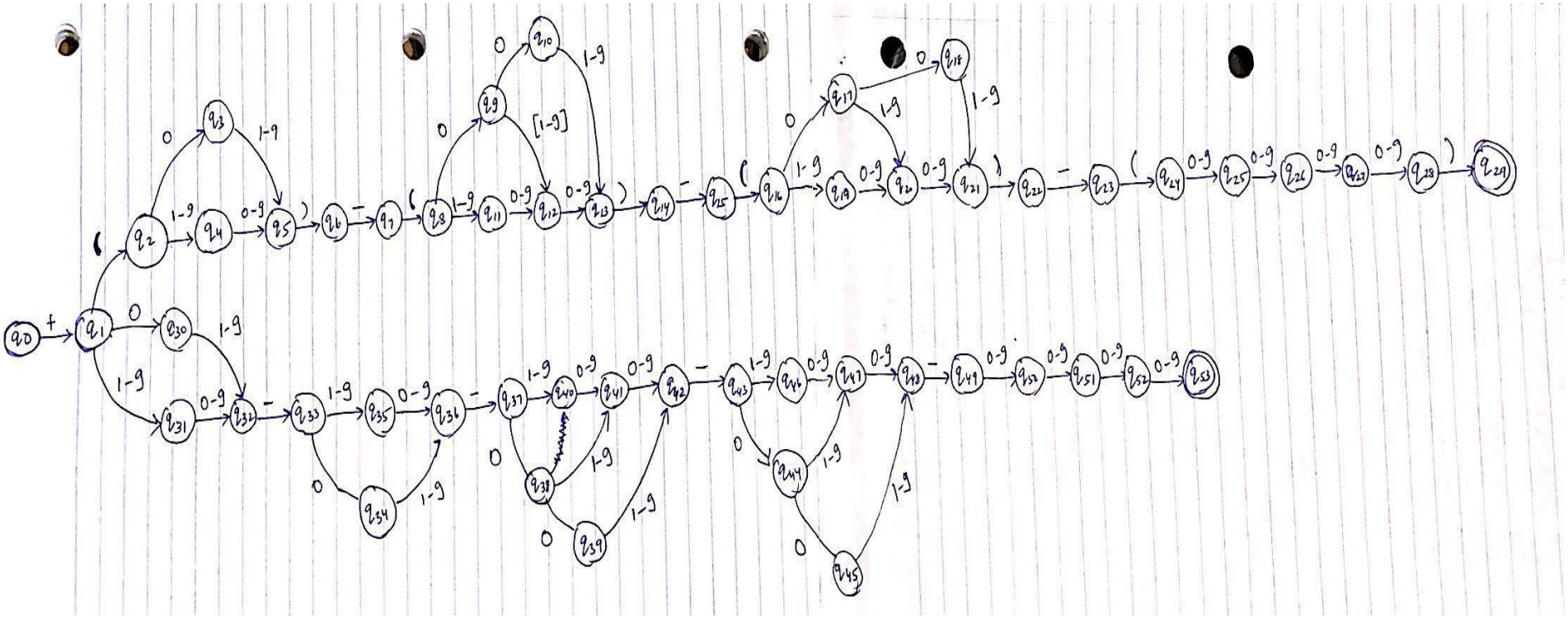
2. Telephone Number (50 points)

([+][(](?!00)[0-9]{2}[)]-[(](?!000)[0-9]{3}[)]-[(](?!000)[0-9]{3}[)]-[(][0-9]{4}[)])|([+](?!00)[0-9]{2}-(?!000)[0-9]{3}-(?!000)[0-9]{3}-[0-9]{4})

Snapshot from regexr.com:



Deterministic FSA attached below.


q0
+
q1
0
1-9
q30
q31
0-9
q32
-
q33
1-9
q35
0-9
q36
-
q37
1-9
q40
0-9
q41
0-9
q42
-
q43
1-9
q46
0-9
q47
0-9
q48
-
q49
0-9
q50
0-9
q51
0-9
q52
0-9
q53
0
q34
1-9
0
q38
1-9
0
q39
1-9
0
q44
1-9
0
q45
1-9
(
q2
0
1-9
q3
q4
0-9
q5
)
q6
=
q7
(
q8
1-9
q11
0-9
q12
0-9
q13
)
q14
=
q15
(
q16
0
1-9
q17
q18
1-9
q19
0-9
q20
0-9
q21
)
q22
=
q23
(
q24
0-9
q25
0-9
q26
0-9
q27
0-9
q28
)
q29
0
1-9
[1-9]