CS – 204: Assignment 1

15 . 02 . 2021

Table 1: *Group 13*

Sch. ID	**Name**
1912088	Subham Patar
095	Jason Hrangkhol
165	Aditya Singh
170	Mritunjay Aman
176	Soumen Pradhan

1. $L^* -$

   ab aa baa ab aa
   aa aa baa aa
   baa aa ab aa

   $L^4 -$

   aa aa baa aa
   baa aa ab aa

2. Given,

$$\Sigma = \{a, b\}$$

$$L = \{aa, bb\}$$

So,

$$\begin{aligned} \overline{L} &= \Sigma^* - L \\ &= \{w \mid w \in \Sigma^n, n > 2\} + \{\lambda, a, b, ab, ba\} \end{aligned}$$

3. Note: Regex is Regular Expression
   a) Regex - $b * ab*$
      $P : S \to bS \mid Sb \mid a$
      $G = (\ \{S\}, \{a, b\}, S, P\ )$

   b) Regex - $b * ab * a * b*$
      $P : S \to aS \mid bS \mid Sb \mid a$
      $G = (\ \{S\}, \{a, b\}, S, P\ )$

   c) Regex - $b * ab * ab * ab*$

$P:$

$S \to aS_1 \mid bS \mid \lambda$
$S_1 \to aS_2 \mid bS_1 \mid \lambda$
$S_2 \to aS_3 \mid bS_2 \mid \lambda$
$S_3 \to bS_3 \mid \lambda$

$G = (\{S, S_1, S_2, S_3\}, \{a, b, \lambda\}, S, P)$

d) Regex - $b * ab * ab * ab * a * b*$

$P:$

$S \to aS_1 \mid bS$
$S_1 \to aS_2 \mid bS_1$
$S_2 \to aS_3 \mid bS_2$
$S_3 \to aS_3 \mid bS_3 \mid \lambda$

$G = (\{S, S_1, S_2, S_3\}, \{a, b, \lambda\}, S, P)$

4. a) Regex - $(ab) * b$

$P : S \to aSb \mid Sb \mid b$

$G = (\{S\}, \{a, b\}, S, P)$

b) Regex - $a * (bb)*$

$P : S \to aSbb \mid \lambda$

$G = (\{S\}, \{a, b\}, S, P)$

c) Regex - $(aaa)(ab) * b$

$P : S \to aSb \mid aa$

$G = (\{S\}, \{a, b\}, S, P)$

d) Regex - $(aaa)(ab)*$

$P : S \to aSb \mid aaa$

$G = (\{S\}, \{a, b\}, S, P)$

e) Regex - $(ab) * ba * (bb)*$

$P:$

$S \to S_1 S_2 B$
$S_1 \to aS_1 b \mid S_1 b \mid b$
$S_2 \to aS_2 bb \mid \lambda$

$G = (\{S, S_1, S_2\}, \{a, b, \lambda\}, S, P)$

f) Regex - $(ab) * b + a * (bb)*$

It can be observed that $L_1 \cup L_2 = L_1 \cup \{\lambda\}$.

$P : S \to aSb \mid Sb \mid b \mid \lambda$

$G = (\{S\}, \{a, b, \lambda\}, S, P)$

g) Regex - $(ab) * b(ab) * b(ab) * b$

$P:$

$S \to SSS$

$S \to aSb \mid Sb \mid b$

$G = (\{S\}, \{a, b\}, S, P)$

h) Regex - $((ab) * b)*$

$P:$

$S \to AB$

$A \to aAb \mid Ab \mid b$

$B \to AB \mid \lambda$

$G = (\{S, A, B\}, \{a, b, \lambda\}, S, P)$

5. We will prove that the 2 grammars, G_1 and G_2 are not equivalent by providing a counterexample.

   Deriving from the G_1,

$$S \Rightarrow aSb \Rightarrow aSSb \Rightarrow aaSb \Rightarrow aaab$$

   Similarly, for G_2

$$S \Rightarrow aSb \Rightarrow aab$$
$$\Rightarrow aaSbb \Rightarrow aaabb$$

   Thus, $aaab \in G_1$ and $aaab \notin G_2$. As a result, they are not equivalent.

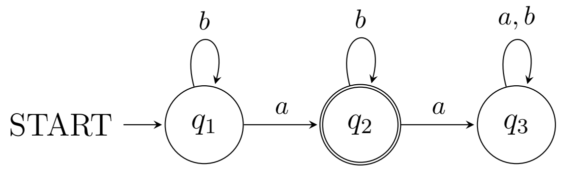

Figure 1: *All strings with exactly one a*

6. a)

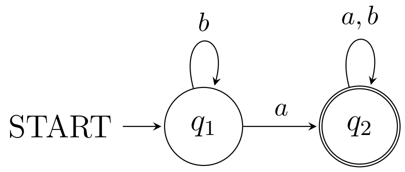

Figure 2: *All strings with at least one a*

   b)

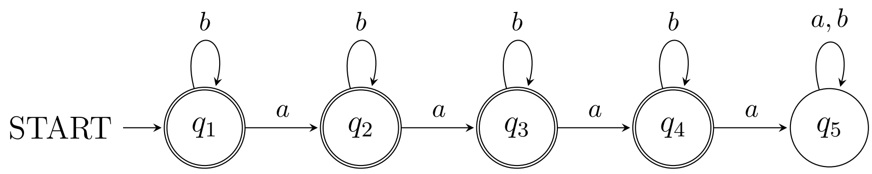

Figure 3: *All strings with no more than 3 a's*

c)

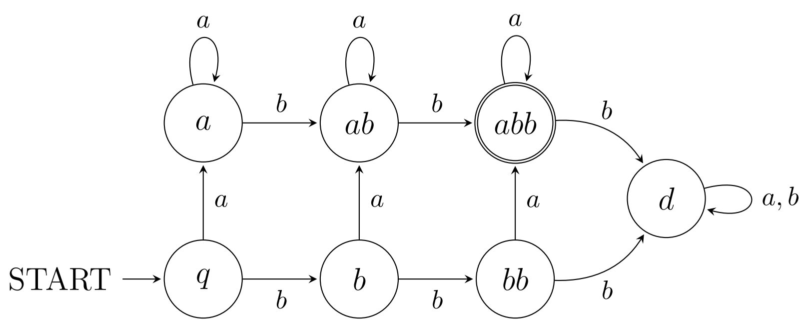

Figure 4: *All strings with at least one a and exactly two b's*

d)

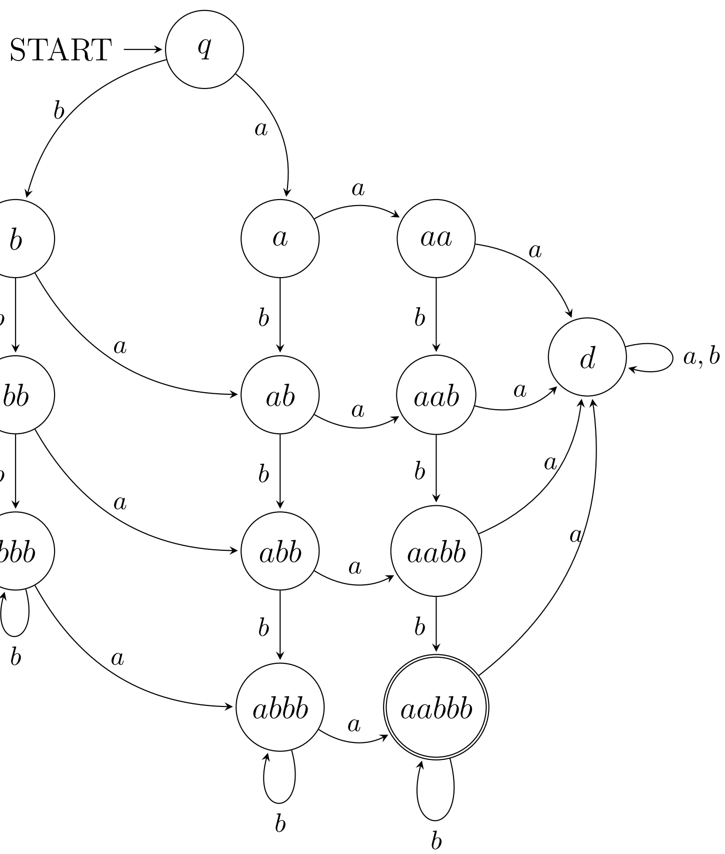

Figure 5: *All strings with exactly two a's and more than two b's*

e)

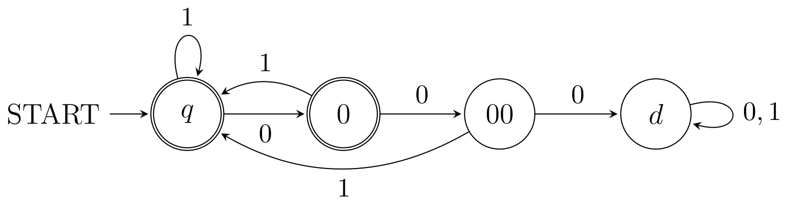

Figure 6: *All strings where every 00 is followed by 1*

7. a)

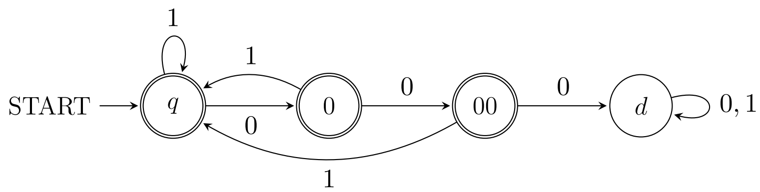

Figure 7: *All strings containing 00 but not 000*

b)

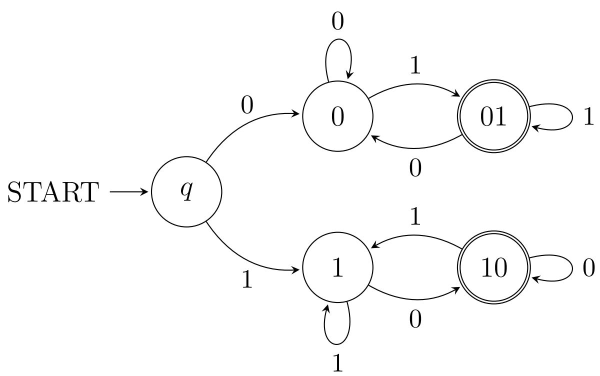

Figure 8: *All strings where leftmost symbol differs from rightmost*

c)

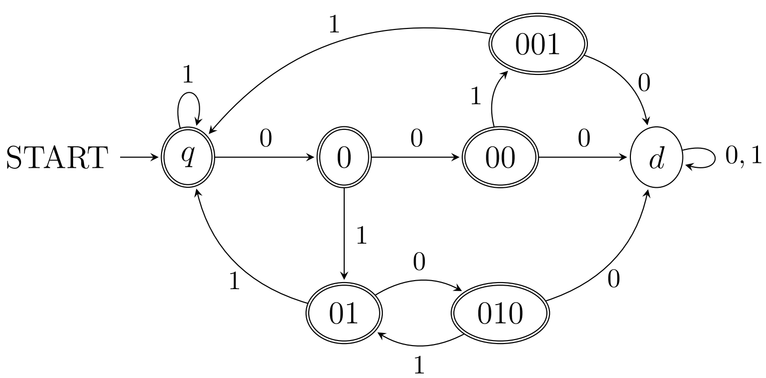

Figure 9: *All strings where every substring of 4 symbols has at most 2 0's*

d)

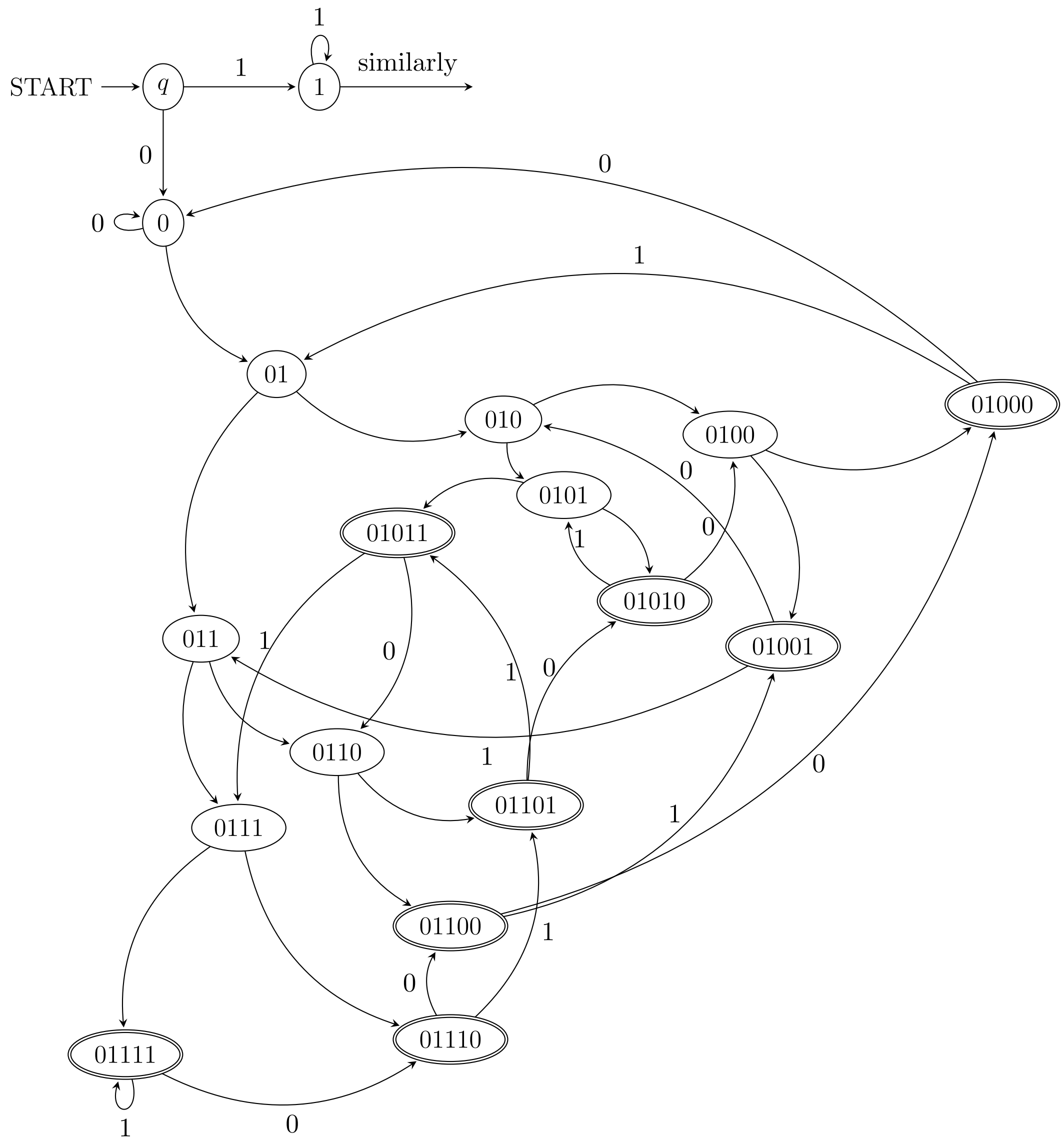

Figure 10: *All strings of length 5 or more where 4th symbol from the right is different from leftmost*

e)

Note that only 0's branch of the dfa is drawn. The 1's branch can generated in a similar way. Also, the transitions that are clearly marked by the states itself are ommitted to avoid cluttering.

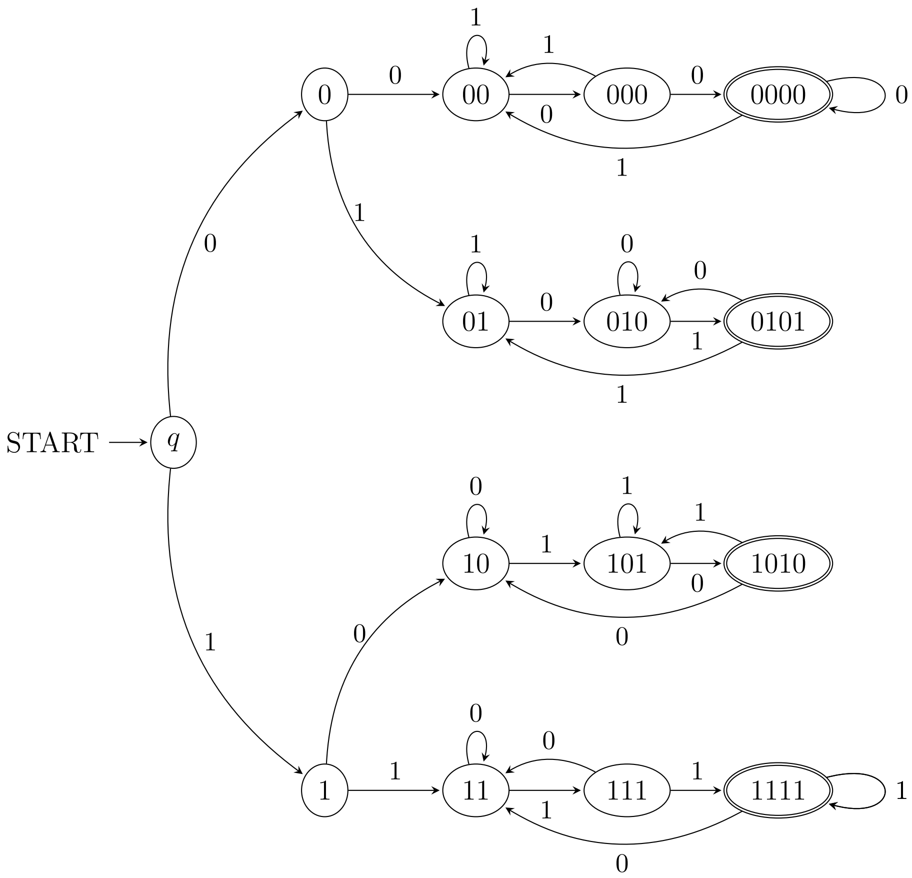

Figure 11: *All strings where leftmost two and rightmost two symbols are identical*

f)

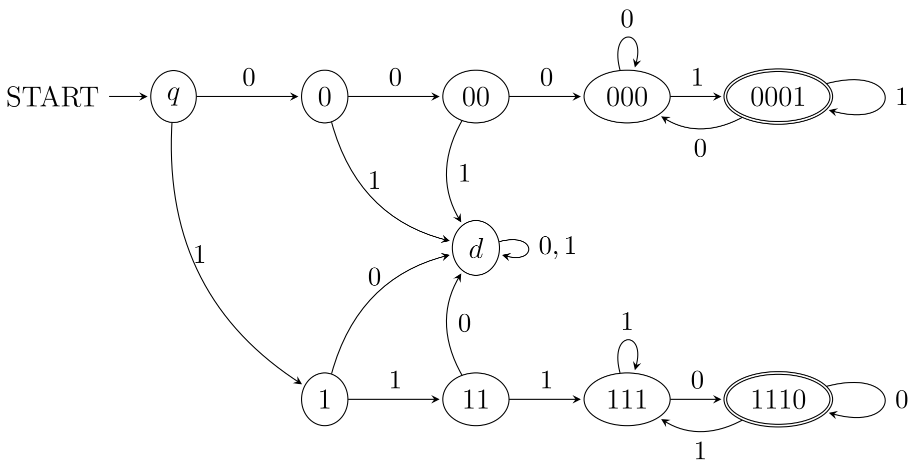

Figure 12: *All strings where leftmost 3 symbols are identical, but different from rightmost symbol*

g)