Theory of Automata - HW3

Vinh T. Nguyen. Email: vinh.nguyen@ttu.edu

October 7, 2018

1 Question 1

Find all strings in *L((a+b)*b(a+ab)*)* of length four.

Answers: We can denote the set of strings as w_1bw_2 where $|w_1|+|w_2| = 3$ and $0 \leq |w_1|, |w_2| \leq 3$

Case 1: $|w_1| = 3, |w_2| = 0$. When the machine (M) consumes λ, thus $|w_2| = 0$. Since w_1 can take any values in $\{$ a,b$\}$. So we have at most $2^3 = 8$ combinations. L1 = $\{$ (aaa)$b\lambda$, (aab)$b\lambda$, (aba)$b\lambda$, (abb)$b\lambda$, (baa)$b\lambda$, (bab)$b\lambda$, (bba)$b\lambda$, (bbb)$b\lambda$ $\}$

Case 2: $|w_1| = 2, |w_2| = 1$. This is when M consumes only one a. Since w_1 can take any values in $\{$ a,b$\}$. So we have at most $2^2 = 4$ strings. L2 = $\{$ (aa)ba, (ab)ba, (ba)ba, (bb)ba $\}$

Case 3: $|w_1| = 1, |w_2| = 2$. When $|w_2| = 2$ we have w_2 = $\{$ aλa, ab$\}$ or $\{$ aa, ab$\}$. Since $|w_1| = 1$ so w_1 can be either a or b. We have $2 * 2 = 4$ strings. L3 = $\{$ (a)b(aa), (b)b(aa), (a)b(ab), (b)b(ab) $\}$

Case 4: $|w_1| = 0, |w_2| = 3$. When $|w_2| = 3$ we have L4 = $\{$ baλaλa, b(ab)λa, baλ(ab)$\}$ or $\{$ baaa, baba, baab$\}$

Final strings by set union of four Ls: $L = L1 \cup L2 \cup L3 \cup L4$ = $\{$aaab, aabb, abab, abbb, baab, babb, bbab, bbbb, aaba, abba, baba, bbba, abaa, bbaa, baaa$\}$

2 Question 2

Find a regular expression for the set $\{a^n b^m | (n+m)$ is even $\}$

There are two cases where $n+m$ is even

Case 1: n, m are both even then we can denote $n = 2k, m = 2l$ and $m, n \geq 0$. The regular expression for this case is $(aa)^*(bb)^*$.

Case 2: n, m are both odd then we can denote $n = 2k+1, m = 2l+1$ and $m, n \geq 0$. The regular expression for this case is $a(aa)^*b(bb)^*$.

We can build a finite machine M that accepts either **Case 1** or **Case 2**. Then the regular expression for this set is: $r = (aa)^*(bb)^* + a(aa)^*b(bb)^*$

3 Question 5

Give a regular expression for a language on $\sum = \{a, b, c\}$ that all strings containing no more than three $a's$.

There are three cases when strings contain no more than three $a's$.

Case 1: When the strings contain no a. The machine M can take any string of $\{$ b, c$\}$ to go to final state. Thus the regular expression for this case is $(b+c)^*$

Case 2: When the strings contain one a. The machine M can take any string of $\{$ b, c$\}$ with one a (or one a and any $\{$ b, c$\}$ to go to final state. Thus the regular expression for this case is $(b+c)^*a(b+c)^*$

Case 3: When the strings contain two $a's$. The machine M can take any string of $\{$ b, c$\}$ with two $a's$ in any combination to go to final state. Thus the regular expression for this case is $(b+c)^*a(b+c)^*a(b+c)^*a(b+c)^*$

We can build a finite machine M that accepts **Case 1, Case 2, or Case 3**. Thus the regular expression is

$r = (b+c)^* + (b+c)^*a(b+c)^* + (b+c)^*a(b+c)^*a(b+c)^*a(b+c)^*$

4 Question 7

Give a regular expression for L on $\sum = \{a, b\}$ where $L = \{w|n_a(w)$ mode 3 =0$\}$

There are two cases when $n_a(w)$ mod 3 = 0.

Case 1: When the strings contain no a or $n_a(w) = 0$. In this case, the regular expression is b^*.

Case 2: When the strings contain k times number of 3 $a's$ (mod 3 of $n_a(w)$ = 0). The regular expression for strings is $(b^*ab^*ab^*ab^*)^*$.

We can build a finite machine M that accepts **Case 1, Case 2**. Thus the regular expression is $r = b^* + (b^*ab^*ab^*ab^*)^*$

5 Question 8

Find a dfa that accepts $L(aa^* + aba^*b^*)$. The procedure is as follows: (1) Find nfa for aa^*, (2) Find nfa for aba^*b^*, (3) Build a transition table for (1) and (2), (4) Build equivalent dfa from transition table, (5) Minimal dfa. The final dfa is shown in Fig. 1

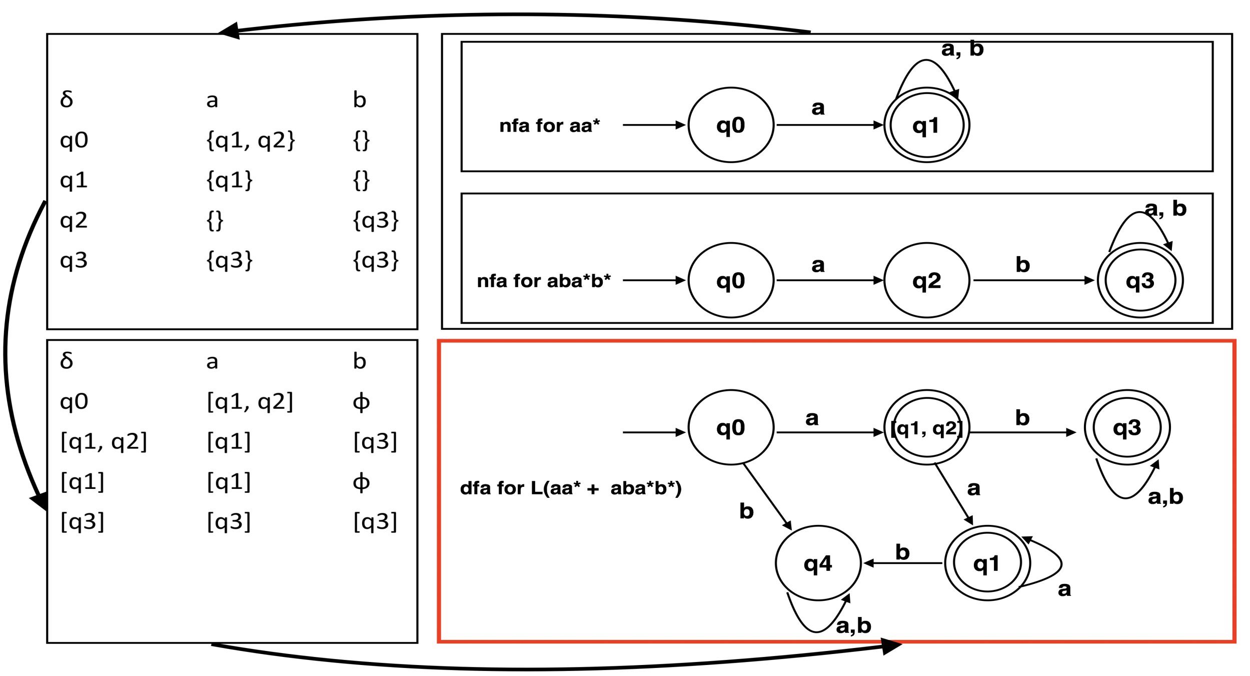

Figure 1: dfa that accepts $L(aa^* + aba^*b^*)$.

6 Question 10

Find a regular expression for the language accepted by the automaton:

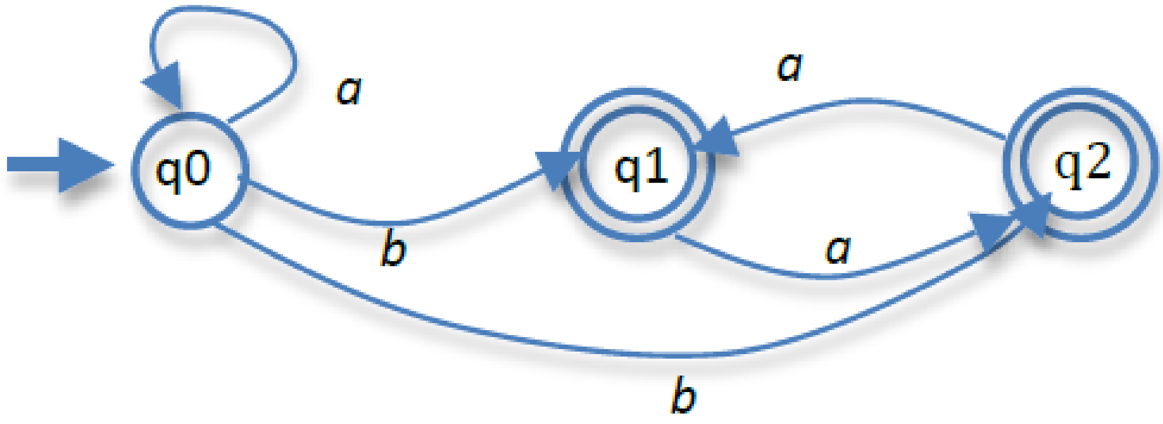

Figure 2: Find a regular expression for the language accepted by the automaton

We have two cases:

Case 1 for q1: we have two regular expressions: $a^*b(aa)^*$ and $a^*ba(aa)^*$

Case 1 for q2: we have two regular expressions: $a^*ba(aa)^*$ and $a^*b(aa)^*$

Set union of **Case 1 and Case 2** on $q1, q2$ we have $r = a^*b(aa)^* + a^*ba(aa)^* = a^*ba^*$. Since M accepts both odd and even number of a at the end.

Another approach is to find minimal automaton as shown in Fig.3

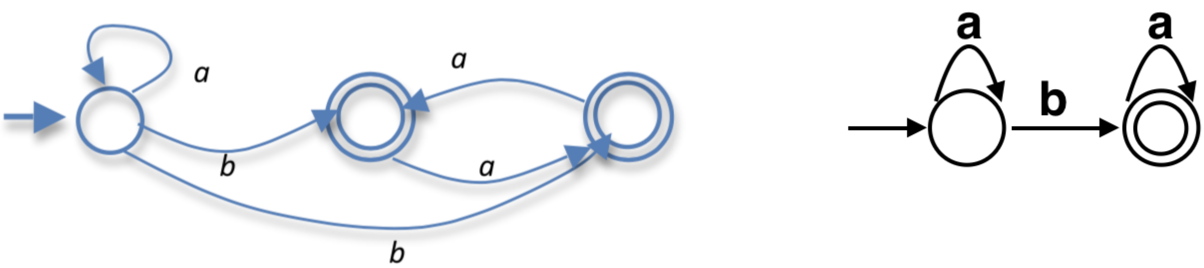

Figure 3: Minimal nfa, $r = a^*ba^*$

7 Question 11

Write a regular expression for the set of all C real numbers. The nfa for set of all C real numbers can be shown in Fig. 4 There are three cases for reaching final states from q0.

Case 1: From $q0$ to $q2$. We have the regular expression: $r = (``+" + ``-" + \lambda)0$

Case 2: From $q0$ to $q4$. We have the regular expression: $r = (``+" + ``-" + \lambda)(1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)^*$

Case 3: From $q0$ to $q3$. We have two paths: go through $q4$ and go through $q2$.

From $q0$ *to* $q3$ *through* $q4$ then we have the following expression: $r = (``+" + ``-" + \lambda)(1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)^*(``.")(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)^*$.

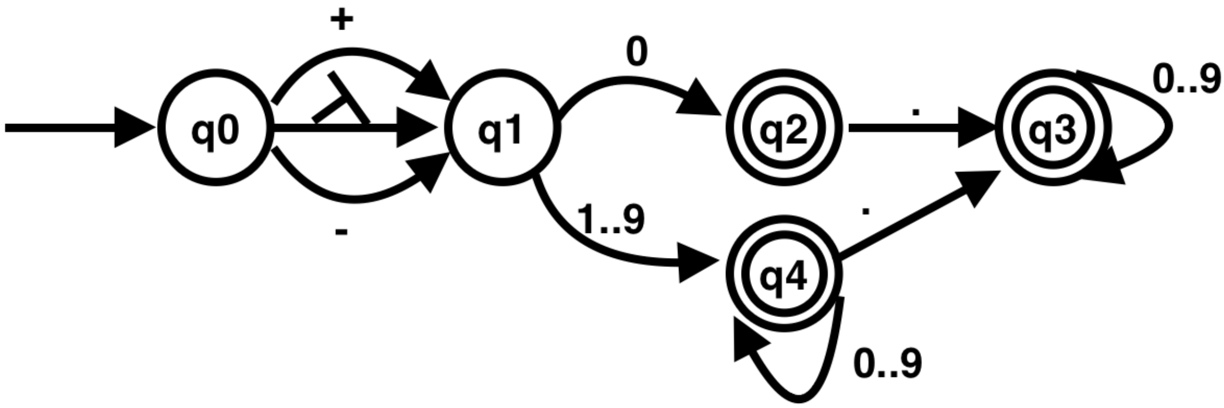

Figure 4: Nfa for C real numbers

From $q0$ *to* $q3$ *through* $q2$ then we have the following expression: $r = (``+'' + ``-'' + \lambda)0(``.'')(0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9)(0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9)^*$

Final regular expression is: $r = (``+'' + ``-'' + \lambda)0 + (``+'' + ``-'' + \lambda)(1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)^* + (``+'' + ``-'' + \lambda)(1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)^*(``.'')(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)^* + (``+'' + ``-'' + \lambda)0(``.'')(0+1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)^*$

8 Question 13

Construct right and left linear grammars for the language $L = \{a^n b^m | n \geq 2, m \geq 3\}$

The right linear grammar $G_R = (\{S, S1, S2, S3\}, \{a, b\}, S, P\})$ with productions P: $S \to aaS1$, $S1 \to aS1|S2$, $S2 \to bbbS3$, $S3 \to bS3|\lambda$

The left linear grammar $G_L = (\{S, S1, S2, S3\}, \{a, b\}, S, P\})$ with productions P: $S \to S1bbb$, $S1 \to S1b|S2$, $S2 \to S3aa$, $S3 \to S2a|\lambda$

9 Question 15

Find a regular grammar that generates the set of all C real numbers. Based on what we've constructed for regular expression in Fig. 4. We can construct regular grammar as follows:

$G = (\{S, S1, S2, S3, S4\}, \{+, -, ., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, S, P\})$
$S \to \lambda S1| + S1| - S1$
$S1 \to 0S2|(1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9)S4$
$S2 \to .S3|\lambda$
$S3 \to (0 + 1 + 2 + 3 + +4 + 5 + 6 + 7 + 8 + 9)S3|\lambda$
$S4 \to (0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9)S4|.S3|\lambda$