哈工大 2007 年 秋 季 学期

形 式 语 言 试题

学号	
姓名	

题号	一	二	三	四	五	六	七	八	九	总分
分数										

注意行为规范 遵守考场纪律

1. [15 points] True or false(√ for true and × for false)

(1) A DFA with n states must accept at least one string of length greater than n. ×

(2) If F is a finite language and L is some language, and L-F is a regular language, then L must be a regular language. √

(3) Define FOUR(w), for a finite string w, to be the string consisting of the symbols of w in positions that are multiples of four. For example, FOUR(1110011100) = 01.

If L is a regular language, then { FOUR(w) | $w \in L$ } must be regular. √

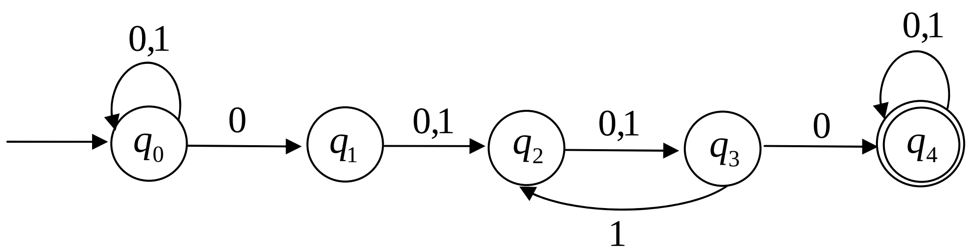
(4) For every three regular expressions R, S, and T, the languages denoted by $R(S+T)$ and $(RS)+(RT)$ are the same. √

(5) If L_1 and L_2 are languages such that L_2, L_1L_2 are all regular, then L_1 must be regular.×

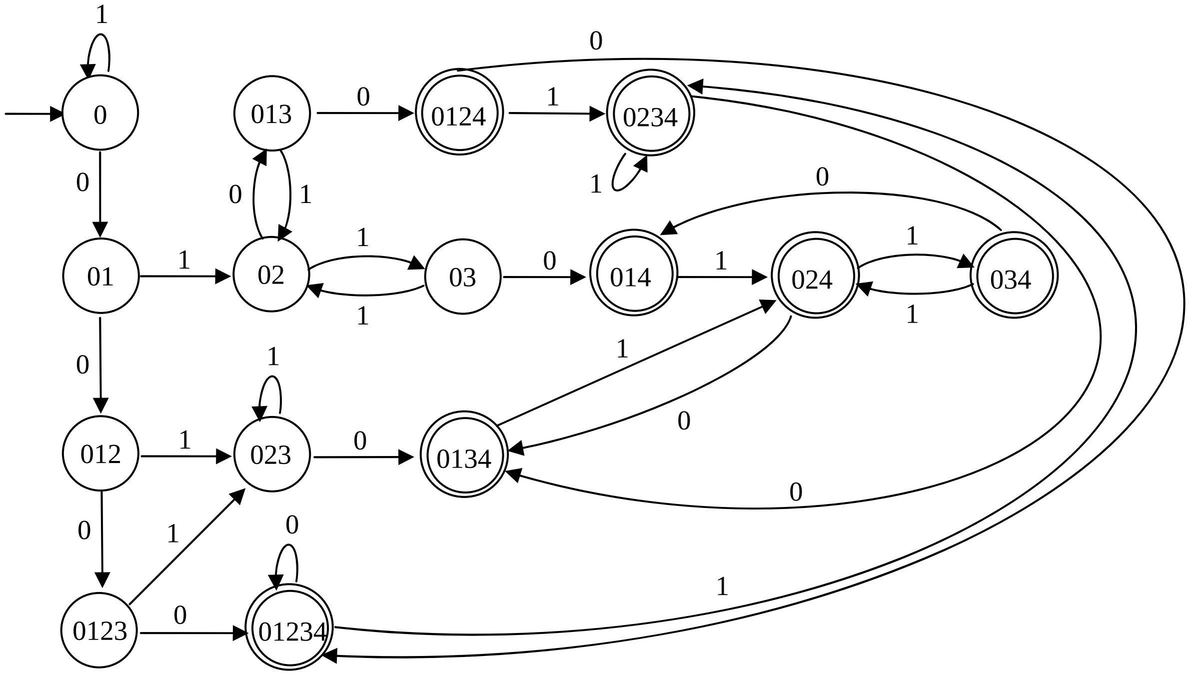
2. [15 points] Let $L \subset \Sigma^*$ be the language of all strings such that there are two 0's separated by a number of positions that is a non-zero multiple of 2 . For example, 0101110 is not in L, but 1011110 and 101001 are in L .

(1) [7 points] Construct an NFA for this language.

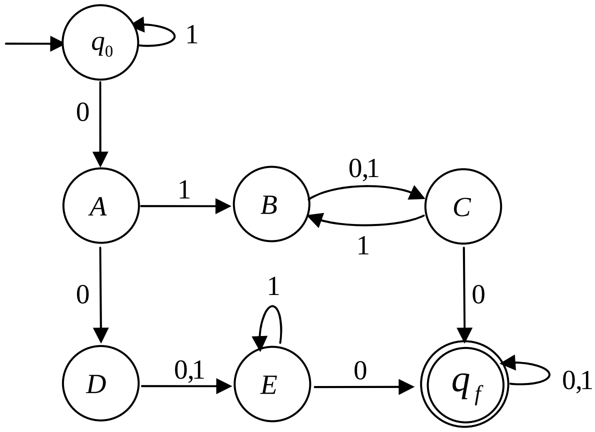
主管领导审核签字



(2) [5 points] Convert this NFA into DFA, using sub-set construction.(Your answer should be the state diagram of a DFA. Your diagram should include only the states that are reachable from start state. Please label your states in some meaningful way.)



(3) [3 points] Give the minimization of this DFA.



3. [10 points] Consider an NFA $N_1 = (Q, \Sigma, \delta, p, F_1)$ with language $L_1 = L(N_1)$. Define a new NFA $N_2 = (Q, \Sigma, \delta, p, F_2)$ with the set of final states $F_2 = Q-F_1$.

Prove that the language $L_2 = L(N_2)$ is the complement of the language L_1, i.e., $L_2 = \Sigma^*-L_1$.

Proof :

Define δ^* as follows :

$\delta^*(p,\varepsilon)=p$

$\delta^*(p,xa)= \delta(\delta^*(p,x),a)$, for any $x\in\Sigma^*$, and any $a\in\Sigma$.

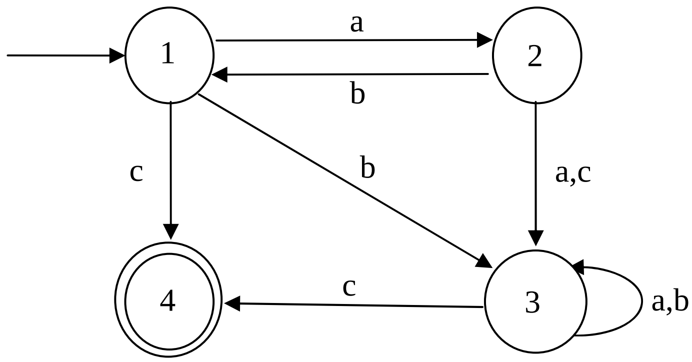
Then $L_2=L(N_2)=\{ w \mid \delta^*(p,w) \in F_2 \}=\{ w \mid \delta^*(p,w) \in Q-F_1 \}$

If $w\in L_2$, then $\delta^*(p,w) \in F_2 = Q-F_1$. That is $w\notin L_1$. So $w\in\Sigma^*-L_1$.

If $w\in\Sigma^*-L_1$, then $w\notin L_1$. That is $\delta^*(p,w) \in Q-F_1 = F_2$. So $w\in L_2$.

Conclusion, $L_2 = L(N_2)$ is the complement of the language L_1, i.e., $L_2 = \Sigma^*-L_1$.

4. [10 points] Consider the DFA $M = (Q, \Sigma, \delta, q_0, F)$ with $Q=\{1, 2, 3, 4 \}$, $Q=\{ a, b, c \}$, $q_0= 1$, $F= \{ 4 \}$, and δ as defined in the following transition diagram.



(1) [6 points] For the DFA M given above, specify the following regular expressions. You do not need to justify your answer and are free to use any method to determine the answer (including “reasoning it out").

a) $R_{11}^{(0)} = \varepsilon$

b) $R_{12}^{(0)}$ =a

c) $R_{33}^{(0)}$ =a+b

d) $R_{12}^{(4)}$ =a(ba)*

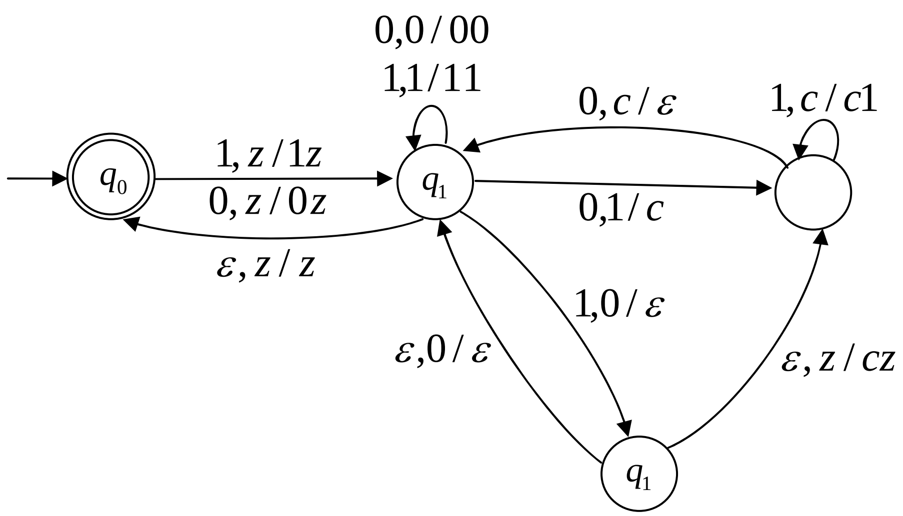
e) $R_{13}^{(4)}$ =(b+a(ba)*(a+c))(a+b)*

(2) [4 points] Provide a regular expression for L(M).

(b+a(ba)*(a+c))(a+b)*c

Or : (b+a(ba)*(a+c))(a+b)*(a+b) (according to the original diagram)

5. [10 points] Let L be a set of all strings of 0's and 1's, such that all the strings contain twice as many 0's as 1's. Give a diagram of the deterministic PDA for L .



6. [10 points]

(1) [3 points] Give the pumping lemma for regular languages.

Let *L* be a regular language. Then there exist a constant n such that for every string w in *L* such that $|w| \geq n$, we can break w into three strings, w=xyz, such that :

a. $y \neq \varepsilon$

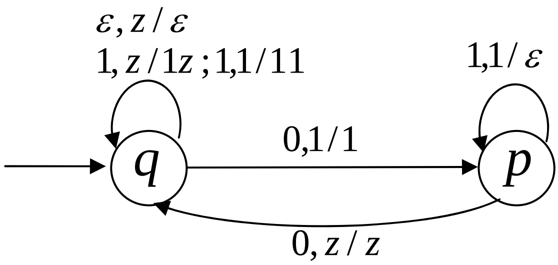
b. $|xy| \leq n$

c. $xy^kz \in L$, for all $k \geq 0$

(2) [7 points] Prove the pumping lemma for regular languages.

See text book : page 127

7. [10 points] A PDA is as following.



(1) [7 points] Convert this PDA into CFG(taking [qxp] as variable)

δ(q,1,z)=(q,1z) ⇒ [qzq] → 1[q1q][qzq] | 1[q1p][pzq]

[qzp] → 1[q1q][qzp] | 1[q1p][pzp]

δ(q,1,1)=(q,11) ⇒ [q1q] → 1[q1q][q1q] | 1[q1p][p1q]

[q1p] → 1[q1q][q1p] | 1[q1p][p1p]

δ(q,ε,z)=(q, ε) ⇒ [qzq] → ε

δ(q,0,1)=(p,1) ⇒ [q1q] → 0[p1q]

[q1p] → 0[p1p]

δ(p,1,1)=(p, ε) ⇒ [p1p] → 1

δ(p,0,z)=(q, z) ⇒ [pzq] → 0[qzq]

[pzp] → 0[qzp]

(2) [3 points] Simplify the CFG you have just got in part (1).

[qzq] → 1[q1p][pzq] | ε

[q1p] → 1[q1p][p1p]

[q1p] → 0[p1p]

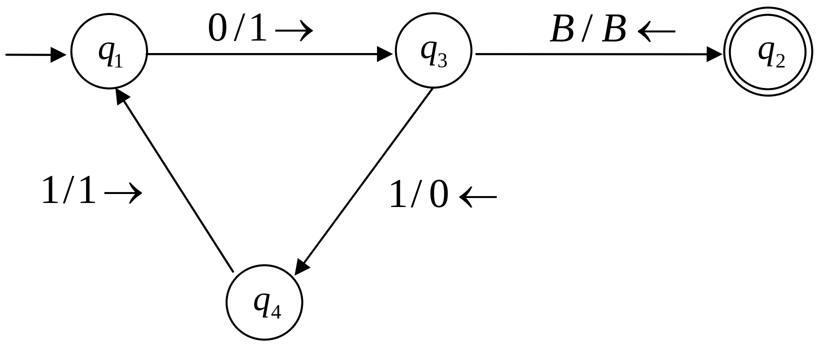
[p1p] → 1

[pzq] → 0[qzq]

8. [10 points] Let following string be a code for Turing machine M ,

0101000100100110001001000010101100001001010010011000100010010001 0 .

(1) [5 points] Give the diagram of M



(2) [5 points] Describe the language of M

L(M)={0}{1}*

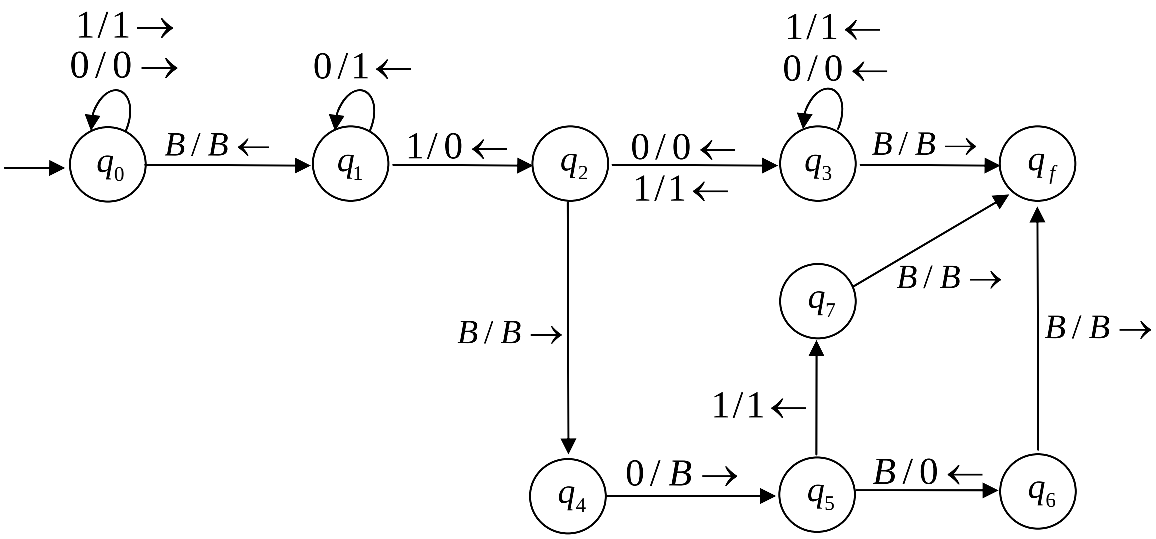
9. [10 points] You are required to construct a Turing machine which takes as input an integer n(n≥1) in binary and subtracts 1 from it. The tape initially contains the number n in binary. The tape head is initially scanning the leftmost bit of n. Your machine should halt with the binary number n-1 on the tape and the head scanning the leftmost bit again. It is also required that you replace any leading zeroes by the B symbol. We give below the initial and final configurations for some sample inputs using the initial state q_0 and halting state q_f .

$q_0 100101 \overset{*}{\Rightarrow} q_f 100100$

$q_0 100000 \overset{*}{\Rightarrow} q_f 11111$

You may assume that the input string is always in the specified format and does not contain any leading zeroes.

(1) [7 points] Give the transitions diagram for this Turing machine.



(2) [3 points] Provide ID's of your machine for the input 1010.

$q_0 1010 \Rightarrow 1q_0 010 \Rightarrow 10q_0 10 \Rightarrow 101q_0 0 \Rightarrow 1010q_0 B \Rightarrow 101q_1 0 \Rightarrow 10q_1 11 \Rightarrow 1q_2 001$
$\Rightarrow q_3 1001 \Rightarrow q_3 B1001 \Rightarrow q_f 1001$