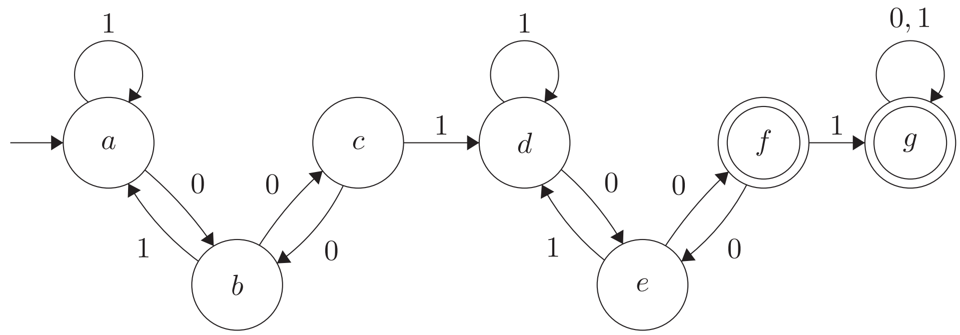
CS/ECE 374 Spring 2023 William Cheng (shihuac2@illinois.edu)

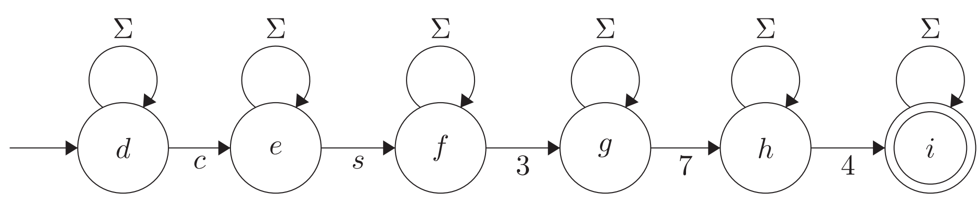
Homework 3 Problem 1

Solution: (a) The NFA is shown below.



- a: have not seen a zero yet
- b: the current length of 0 block is odd
- c: the current length of 0 block is even
- d: one block of 0's of even length has been seen, look for the second 0 block
- e: the current length of 0 block is odd
- f: the current length of 0 block is even
- g: exactly two blocks of 0's of even length has been seen

(b) The NFA is shown below.

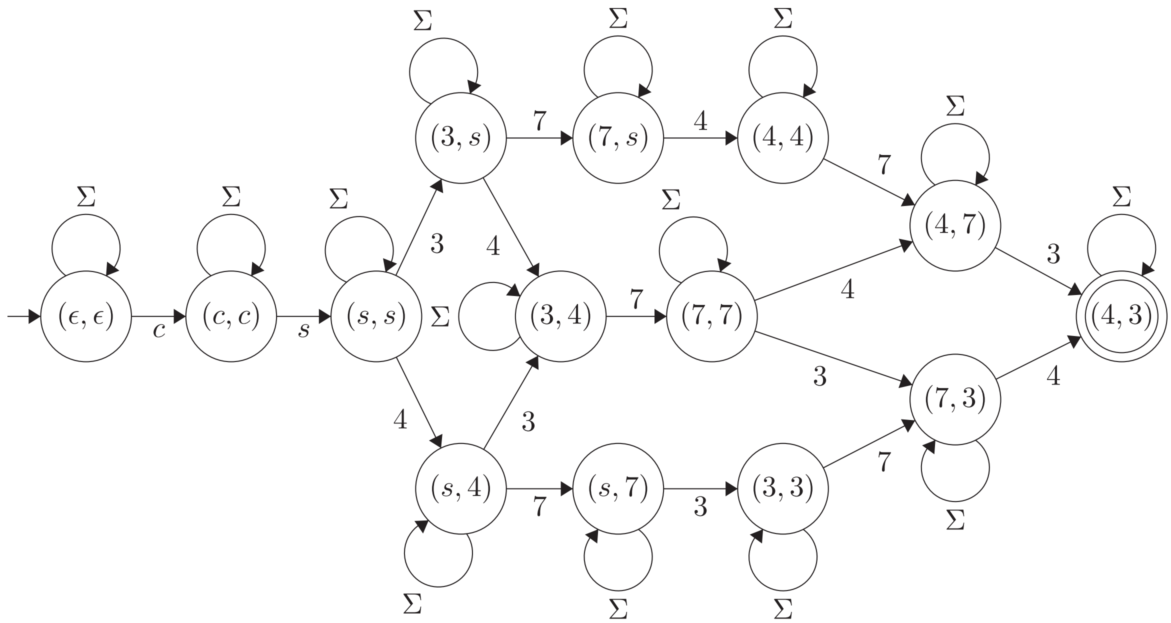


The NFA stays at the current state upon seeing any letter until it sees the next letter in $cs374$, starting from the first letter c. It will stay at the accepting state i after seeing the whole sequence $cs374$.

(c) The states $(q_1, q_2) \in Q_1 \times Q_2$ are defined as follows: Each of q_1 or q_2 is in the set $\{\epsilon, c, s, 3, 7, 4\}$. The state (q_1, q_2) represents that q_1 has been seen in the sequence $cs374$ and q_2 has been seen in the sequence $cs473$.

- $$\delta = \begin{cases} ((\epsilon,\epsilon),\Sigma) \to (\epsilon,\epsilon) \\ ((\epsilon,\epsilon),c) \to (c,c) \\ ((c,c),\Sigma) \to (c,c) \\ ((c,c),s) \to (s,s) \\ ((s,s),\Sigma) \to (s,s) \\ ((s,s),3) \to (3,s) \\ ((s,s),4) \to (s,4) \\ ((3,s),\Sigma) \to (3,s) \\ ((3,s),4) \to (3,4) \\ ((3,s),7) \to (7,s) \\ ((s,4),\Sigma) \to (s,4) \\ ((s,4),3) \to (3,4) \\ ((s,4),7) \to (s,7) \\ ((3,4),\Sigma) \to (3,4) \\ ((3,4),7) \to (7,7) \\ ((7,s),\Sigma) \to (7,s) \\ ((7,s),4) \to (4,4) \\ ((s,7),\Sigma) \to (s,7) \\ ((s,7),3) \to (3,3) \\ ((7,7),\Sigma) \to (7,7) \\ ((7,7),3) \to (7,3) \\ ((7,7),4) \to (4,7) \\ ((4,4),\Sigma) \to (4,4) \\ ((4,4),7) \to (4,7) \\ ((3,3),\Sigma) \to (3,3) \\ ((3,3),7) \to (7,3) \\ ((7,3),\Sigma) \to (7,3) \\ ((7,3),4) \to (4,3) \\ ((4,7),\Sigma) \to (4,7) \\ ((4,7),3) \to (4,3) \\ ((4,3),\Sigma) \to (4,3) \end{cases}$$
- $s = (\epsilon,\epsilon)$
- $A = \{(4,3)\}$

Explanation. For a state (q_1, q_2), if the input letter matches the next letter in $cs374$, q_1 gets updated; if the input letter matches the next letter in $cs473$, q_2 gets updated. Since state $(4,3)$ means that the 4 in $cs374$ has been seen and the 3 in $cs473$ has been seen, it is the only accepting state. Following is a visualization of the NFA.


(ϵ, ϵ)
(c, c)
(s, s)
$(3, s)$
$(7, s)$
$(4, 4)$
$(3, 4)$
$(7, 7)$
$(4, 7)$
$(4, 3)$
$(s, 4)$
$(s, 7)$
$(3, 3)$
$(7, 3)$
c
s
3
4
7
Σ

■

CS/ECE 374 Spring 2023 William Cheng (shihuac2@illinois.edu)
Homework 3 Problem 2

Solution: (a) Let $N = (Q, \Sigma, \delta, s, A)$ be an NFA defined as follows.

- $Q = Q_1 \times Q_2 \times \{\text{before}, \text{in}, \text{after}\}$
- $s = (s_1, s_2, \text{before})$
- $A = \{(q_1, q_2, \text{after}) \mid q_1 \in A_1, q_2 \in A_2\}$
- $\delta((q_1, q_2, \text{before}), a) = \begin{cases} \{(\delta_1(q_1, a), q_2, \text{before}), (q_1, \delta_2(q_2, a), \text{in})\} & \text{if } \delta_2(q_2, a) \neq \emptyset \\ \{(\delta_1(q_1, a), q_2, \text{before})\} & \text{otherwise} \end{cases}$

  $\delta((q_1, q_2, \text{in}), a) = \begin{cases} \{(q_1, \delta_2(q_2, a), \text{in}), (\delta_1(q_1, a), q_2, \text{after})\} & \text{if } \delta_1(q_1, a) \neq \emptyset \\ \{(q_1, \delta_2(q_2, a), \text{in})\} & \text{otherwise} \end{cases}$

  $\delta((q_1, q_2, \text{after}), a) = \{(\delta_1(q_1, a), q_2, \text{after})\}$

Explanation. N non-deterministically chooses a substring w in the input string. It simulates w on M_2, and the rest of the input string on M_1.

- The state $(q_1, q_2, \text{before})$ means N is simulating M_1, the simulation of M_1 is in state q_1, and N has not extracted the substring w yet.
- The state (q_1, q_2, in) means N is simulating M_2, the simulation of M_2 is in state q_2, and N has already extracted the substring w but not finished simulating it on M_2.
- The state (q_1, q_2, after) means N is simulating M_1, the simulation of M_1 is in state q_1, and N has already extracted the substring w and finished simulating it on M_2.

The NFA will reach an accepting state only if there exists a substring w in the input string that is accepted by M_2, and the rest of the input string is accepted by M_1. Therefore, N accepts $\text{insert}(L_1, L_2)$ by definition.

(b) i.
- $r_1 = \emptyset$: $r' = \emptyset$. No strings are in $L(r_1)$ to be inserted.
- $r_1 = \epsilon$: $r' = r_2$, since $L(r') = \{\epsilon w \epsilon | w \in L(r_2)\} = L(r_2)$.
- $r_1 = a$, $a \in \Sigma$: $r' = ar_2 + r_2 a$.

ii. The regular expression for $\text{insert}(L(r_1), L(r_2))$ is $s' + t'$.
Explanation. $L(r_1) = L(s) \cup L(t)$. Any string $xy \in L(r_1)$ is either in $L(s)$ or $L(t)$. Then for some $w \in L(r_2)$, xwy either describes $\text{insert}(L(s), L(r_2))$ or $\text{insert}(L(t), L(r_2))$, therefore $s' + t'$.

iii. The regular expression for $\text{insert}(L(r_1), L(r_2))$ is $s't + st' + sr_2t$.
Explanation. The position for insertion is either in s, or in t, or exactly between s and t.

iv. The regular expression for $\text{insert}(L(r_1), L(r_2))$ is $s^*s's^* + s^*r_2s^*$.
Explanation. The position for insertion is either in one of the s's, or between two blocks (can be empty) of s's.

v. Let $ins(r_1, r_2)$ be the regular expression representing the language $\text{insert}(L(r_1), L(r_2))$.

$$\begin{aligned} &ins(0^* + (01)^* + 011^*0, 101) \\ &= ins(0^*, 101) + ins((01)^*, 101) + ins(011^*0, 101) \\ &= 0^*1010^* + (01)^*101(01)^* + (01)^*01011(01)^* + 101011^*0 + 011^*1011^*0 + 011^*0101 + 010111^*0 \end{aligned}$$

■