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

Homework 2 Problem 1

Solution: (a) $(0+1)^*111(0+1)^*10 + (0+1)^*1110$.

Explanation. There are two cases: 1) the 1 in the ending 10 is a part of the substring 111, which can be represented as $(0+1)^*1110$; 2) the 1 in the ending 10 is not a part of the substring 111, which can be represented as $(0+1)^*111(0+1)^*10$. The desired expression is the union of these two expressions.

(b) $1^*0^*1^*$.

Explanation. Consider a string that does not contain the subsequence 010. Then it contains at most one block of 0's otherwise it must contain the subsequence 010 since there must be at least a 1 between two blocks of 0's. $1^*0^*1^*$ represents all such strings.

(c) $0^*(10^* + 10^*10^*)(10^*10^*10^*)^*$.

Explanation. There are two cases: 1) $\#1's \bmod 3 = 1$; 2) $\#1's \bmod 3 = 2$. The $(10^*10^*10^*)^*$ part accounts for multiples of 3 of 1's. The part before it accounts for both cases, either containing one 1 or two 1's.

(d) $w_1 + w_2 + ... + w_k$.

Explanation. The expression represents the union of all $\{w_i\}$ where $w_i \in L$, which is L itself.

(e) Let $Q = \{x \in \Sigma^* \mid |x| \leq h\} \setminus L = \{q_1, q_2, ..., q_m\}$. Then Q is finite since h is a finite number. The desired expression is:

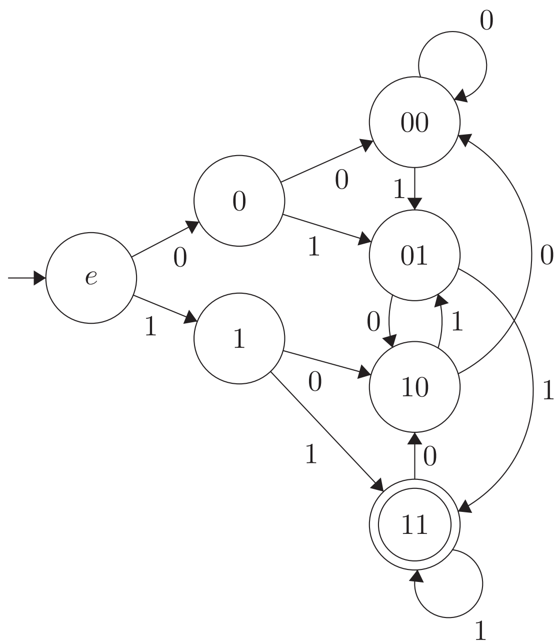
$$q_1 + q_2 + ... + q_m + \underbrace{(0+1)(0+1)...(0+1)}_{h+1}(0+1)^*.$$

Explanation. All strings with length $> h$ are automatically in $\bar{L}$ since the longest string in L has length h. The desired expression is the union of all strings with length $> h$ and all strings with length $\leq$ h that are not in L.

■

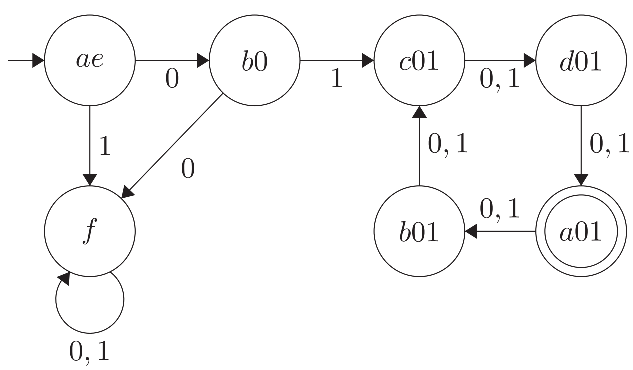
CS/ECE 374 Spring 2023 William Cheng (shihuac2@illinois.edu)
Homework 2 Problem 2

Solution: (a) The DFA is shown below.



The meaning of each state is that the name of each state represents the last two letters of the current longest prefix being read from the input string. Therefore, when the whole input string is read, the state of the DFA represents the last two letters of the input string. The only accepting state is 11.

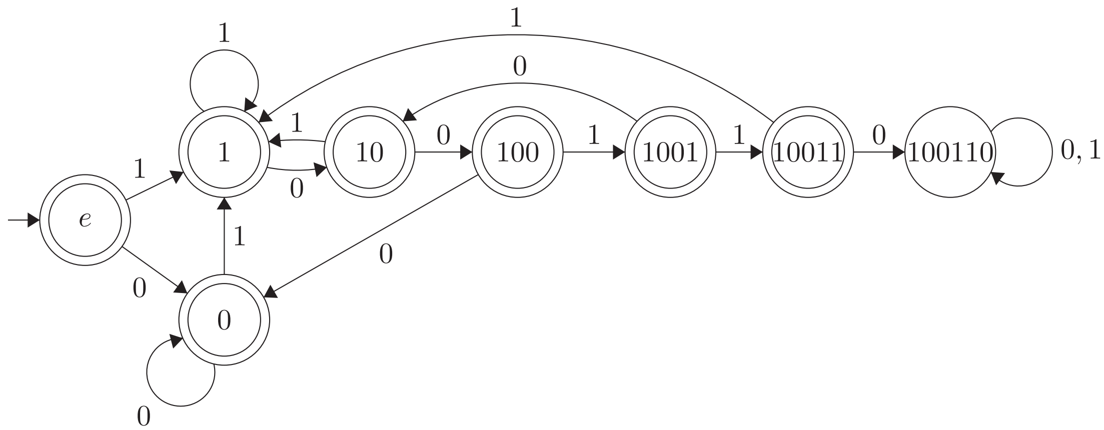
(b) The DFA is shown below.



- ae: the current prefix is ϵ
- $b0$: the current prefix is 0
- f: the input string fails to start with 01
- $c01$: the input string starts with 01, and the current length is 2 mod 4
- $d01$: the input string starts with 01, and the current length is 3 mod 4

- $a01$: the input string starts with 01, and the current length is $0 \bmod 4$
- $b01$: the input string starts with 01, and the current length is $1 \bmod 4$

(c) The DFA is shown below.



The meaning of each state is the last letter(s) of the current longest prefix of the input string. Every state except 100110 is an accepting state, so that when 100110 is a substring, the DFA would not reach an accepting state and the input string will therefore not be accepted. Otherwise, the input string will always be accepted.

(d) $M = (Q, \Sigma, \delta, s, A)$ where

- $Q = \{\epsilon, a_1, a_1a_2, ..., a_1a_2...a_{k-1}, a_1a_2...a_k\}$
- $\Sigma = \{0, 1\}$
- $\delta = (a_1a_2...a_m, a_n) \to \begin{cases} a_1a_2...a_ma_{m+1} & a_n = a_{m+1}, m < k \\ a_1a_2...a_k & m = k \\ a_i...a_ma_n = a_1a_2...a_{m-i} & a_n \neq a_{m+1}, \text{ } i \text{ is the smallest index possible} \\ \epsilon & a_n \neq a_{m+1}, \text{ no suffix matching prefix of } s \end{cases}$

  $(\epsilon, a_n) \to \begin{cases} a_1 & a_n = a_1 \\ \epsilon & a_n \neq a_1 \end{cases}$

- $s = \epsilon$
- $A = \{a_1a_2...a_k\}$

The DFA has $k + 1$ states.

(e) $M = (Q, \Sigma_M, \delta, s, A)$ where

- $Q = \{(q_1, q_2, q_3, q_4) \mid q_1 \in Q_1, q_2 \in Q_2, q_3 \in Q_3, q_4 \in Q_4\}$
- $\Sigma_M = \Sigma$
- $\delta((q_1, q_2, q_3, q_4), a) = (\delta_1(q_1, a), \delta_2(q_2, a), \delta_3(q_3, a), \delta_4(q_4, a))$
- $s = (s_1, s_2, s_3, s_4)$
- $A =$

$$\begin{aligned}\{(a_1, a_2, a_3, a_4) \quad &\mid (a_1 \in A_1 \text{ and } a_2 \notin A_2 \text{ and } a_3 \notin A_3 \text{ and } a_4 \notin A_4) \\ &\text{or } (a_1 \notin A_1 \text{ and } a_2 \in A_2 \text{ and } a_3 \notin A_3 \text{ and } a_4 \notin A_4) \\ &\text{or } (a_1 \notin A_1 \text{ and } a_2 \notin A_2 \text{ and } a_3 \in A_3 \text{ and } a_4 \notin A_4) \\ &\text{or } (a_1 \notin A_1 \text{ and } a_2 \notin A_2 \text{ and } a_3 \notin A_3 \text{ and } a_4 \in A_4)\}\end{aligned}$$

■