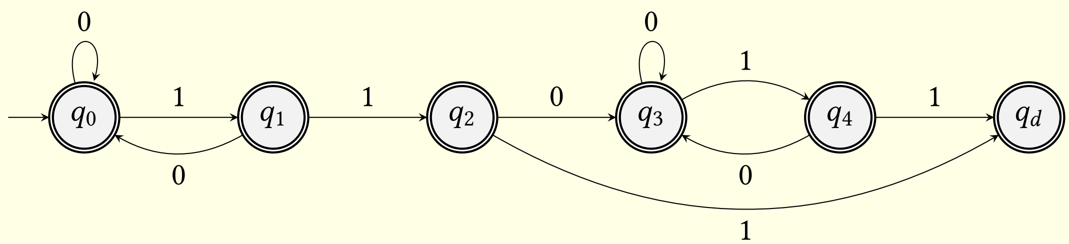
1. Write down regular expressions for the following languages.

(a) $L = \{w \in \{0,1\}^* \mid w \text{ has at most one pair of consecutive } 1s\}$.

Solution: Consider the DFA M accepting L.



By the state elimination method, we get the Regular Expression for L as follows: $(0 + 10)^*(1+\epsilon)(1+\epsilon)((0(0+10)^*(1+\epsilon)) + \epsilon)$.

The order in which states were eliminated for obtaining this expression was q_d, q_0, q_1, q_3, q_4, and q_2.

An alternate way is to use the closure properties. We can write $L = L_1 \cup L_2$, where L_1 is the set of strings that have no pairs of consecutive 1s, and L_2 is the set of strings containing exactly one pair of consecutive 1s.

Now, L_1 can either contain no 1s at all (this corresponds to 0^*), or every occurrence of 1 is either followed by a 0, or 1 is the last symbol of the string (this corresponds to $0^*(100^*)^*(1+\varepsilon)$).

Now, every string $w \in L_2$ can be split in two ways.

- a part that comes from L_1, followed by the 110, followed by a part from L_1, or
- a part that comes from L_1 and ending in 11.

Thus $w \in L_2$ can be written as $0^*(100^*)^*1100^*(100^*)(1+\varepsilon) + 0^*(100^*)^*11$.

Thus, the final regex will be $0^* + 0^*(100^*)^*(1+\varepsilon) + 0^*(100^*)^*1100^*(100^*)(1+\varepsilon) + 0^*(100^*)^*11$.

(b) $L = \{w \in \{0,1\}^* \mid$ the number of zeroes in w is divisible by 3$\}$.

Solution: Consider $R = 1^*(01^*01^*01^*)^*1^*$. If $w \in L(R)$, then all the zeroes appears as multiples of 3, and hence $w \in L$.

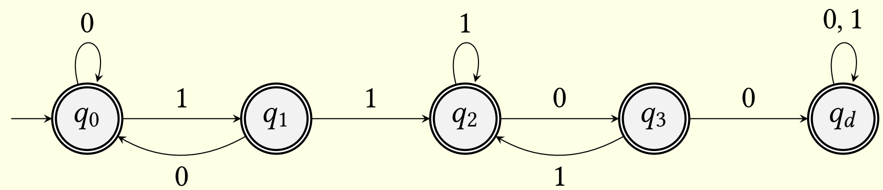
Now, let $w \in L(R)$. Since the number of zeroes in w is divisible by 3, we can partition the string w by grouping together every three closest occurrences of 0. Since these are the closest occurrences, the symbols between them must be 1s.

(c) $L = \{w \in \{0,1\}^* \mid$ every odd position of w is a 1$\}$.

Solution: The first position of the string (from the left) will be a 1. The even positions can be either 0 or 1. The string can be either of even or odd length. Thus, L can be expressed by the regex $R = (1(1+0))^*(1+\epsilon)$.

(d) $L = \{w \in \{0,1\}^* \mid$ every pair of adjacent $0s$ appear before any pair of adjacent $1s\}$.

Solution: Consider the DFA M accepting L.



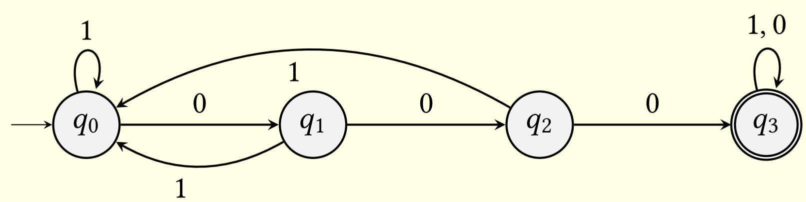
By State Elimination Method, we get the Regular Expression for L as follows: $(0 + 10)^*(\epsilon + 1 + 11)(1 + 01)^*(\epsilon + 0)$

The order in which the states were eliminated for obtaining the expression was q_d, q_0, q_1, q_2, q_3.

2. Describe the languages given by the following regular expressions and construct a DFA accepting the language.

(a) $(0^*1^*)^*000(0+1)^*$.

Solution:

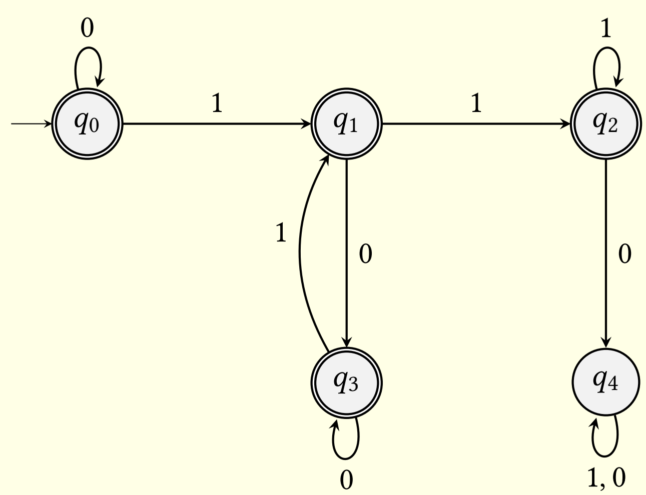


$L = \{w \in \{0,1\}^* \mid w$ contains 000 as a substring $\}$.

Verify the description L matches the regular expression. We have seen a similar example in class.

(b) $(0+10)^*1^*$.

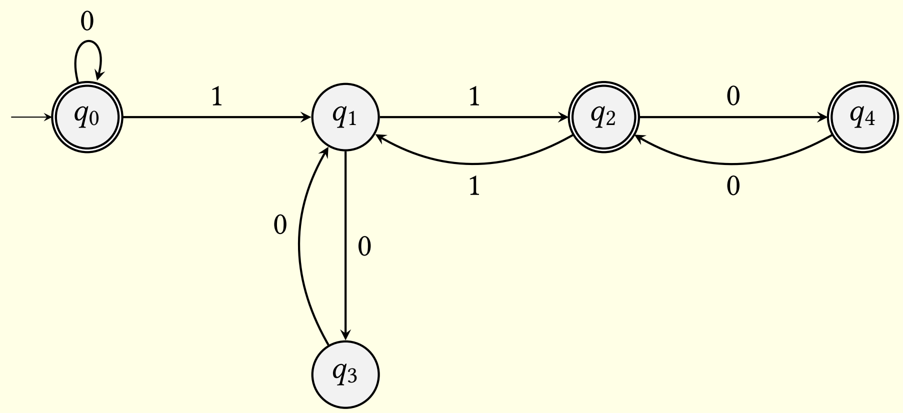
Solution:



$L = \{w \in \{0,1\}^* \mid w$ does not contain a 0 after more than one consecutive 1 $\}$.

(c) $(0+\varepsilon)(00+1(00)^*1)^*(0+\epsilon)$.

Solution:



$L = \{w \in \{0,1\}^* \mid w$ has even number of 1s and number of 0s in between 1s is even $\}$.

Take any $w \in L(R)$. The 1s in the string will always occur in pairs, and the string between them is $(00)^*$ and hence will contain an even number of 0s. Thus $w \in L$.

Take any $w \in L$. If w does not contain 1, then $w = 0^*$. If $|w|$ is odd, then we can take initial 0 from $(0+\varepsilon)$, followed by $(00)^*$ from $(00+1(00)^*1)^*$ and the final ε, and hence $w \in L(R)$.

Otherwise, $w \in L$ and contains a 1. Let $w = w_1w_2w_3$ where w_2 starts and ends with a 1, and w_1, w_3 do not contain 1s. Thus, we have $w_1 \in L((0+\varepsilon)(00)^*)$, and $w_3 \in L((00)^*(0+\varepsilon))$, depending on whether w_1 and w_3 are of even or odd length. Now, $w_2 = w_2'w_2''$, where w_2' starts and ends with a 1 and contains only 0s in between, and w_2'' is remainder of w_2. From the definition of L, $\#0(w_2')$ is even, and hence $w_2' \in L(1(00)^*1)$. Furthermore, w_2'' starts with an even number of 0s as well. Thus $w_2'' \in L((00+1(00)^*1)^*)$. Hence $w_2 = w_2'w_2'' \in L((00+1(00)^*1)^*)$, using the closure properties of regular languages. If $w_1 \in L((0+\varepsilon)(00)^*)$, then $w_1 \in L((0+\varepsilon)(00+1(00)^*1)^*)$ as well. Similarly, $w_3 \in L((00+1(00)^*1)^*(0+\varepsilon))$. Thus $w_1w_2w_3 \in L((0+\varepsilon)(00+1(00)^*1)^*)L((00+1(00)^*1)^*)L((00+1(00)^*1)^*(0+\varepsilon))$, and hence in $L(R)$.

3. Let R be a regular expression over some alphabet Σ, and $L(R) \subseteq \Sigma^*$ denote the language corresponding to it. For a symbol $\sigma \in \Sigma$, the *derivative* of R w.r.t σ, denoted by $\frac{dR}{d\sigma}$ is the language $\{w \mid \sigma w \in L(R)\}$.

(a) Prove that $\frac{dR}{d\sigma}$ is regular.

Solution: Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA corresponding to $L(R)$. Let M' be the DFA $(Q, \Sigma, \delta, q', F)$ where $q' = \delta(q_0, \sigma)$. We will show that $L(M') = \frac{dR}{d\sigma}$.

Suppose that $w \in L(M')$. Then, $\widehat{\delta}(q', w) \in F$. Since $q' = \delta(q_0, \sigma)$, we have $\widehat{\delta}(\delta(q_0, \sigma), w) = \widehat{\delta}(q_0, \sigma w) \in F$, and hence $\sigma w \in L(R)$. Thus, $w \in \frac{dR}{d\sigma}$.

Now, if $w \in \frac{dR}{d\sigma}$, then $\sigma w \in L(R)$. Consequently, $\widehat{\delta}(q_0, \sigma w) \in F$, and hence $\widehat{\delta}(q', w) \in F$. Therefore, $w \in L(M')$.

(b) Suppose that $R = R_1 + R_2$. Prove the following identity.

$$\frac{dR}{d\sigma} = \frac{dR_1}{d\sigma} + \frac{dR_2}{d\sigma}.$$

In other words, the language $\frac{dR}{d\sigma}$ is the union of the languages $\frac{dR_1}{d\sigma}$ and $\frac{dR_2}{d\sigma}$.

Solution: We can write as follows.

$$\begin{aligned} w \in \frac{dR}{d\sigma} &\Leftrightarrow \sigma w \in L(R) \Leftrightarrow \sigma w \in L(R_1 + R_2) \\ &\Leftrightarrow \sigma w \in L(R_1) \text{ or } \sigma w \in L(R_2) \\ &\Leftrightarrow w \in \frac{dR_1}{d\sigma} \text{ or } w \in \frac{dR_2}{d\sigma} \\ &\Leftrightarrow w \in \frac{dR_1}{d\sigma} + \frac{dR_2}{d\sigma}. \end{aligned}$$

(c) Prove the following identity about the derivative of the Kleene closure of R.

$$\frac{d(R^*)}{d\sigma} = \frac{dR}{d\sigma}R^*.$$

In other words, the language $\frac{d(R^*)}{d\sigma}$ is the concatenation of the languages $\frac{dR}{d\sigma}$ and R^*.

Solution: Let $w \in \frac{dR^*}{d\sigma}$. Then $\sigma w \in L(R^*)$. Thus $\sigma w = w_1 \cdot w_2 \cdot w_k$ where $w_i \in L(R)$, and $w_1 = \sigma w_1'$. Thus $w_1' \in \frac{dR}{d\sigma}$, and $w_2 \cdot w_3 \cdots w_k \in L(R^*)$. Thus $w = w_1' \cdot w_2 \cdots w_k \in \frac{dR}{d\sigma}R^*$. This shows that $\frac{dR^*}{d\sigma} \subseteq \frac{dR}{d\sigma}R^*$. The other direction can be argued similarly.

4. Let R_1, R_2, and R_3 be three regexes. Prove the following identities.

(a) $R_1(R_2 + R_3) = R_1R_2 + R_1R_3$.

Solution: We first simplify the LHS. Clearly, $w \in L(R_1(R_2 + R_3)) = L(R_1) \cdot L(R_2 + R_3) \iff w = xy$ for some $x \in L(R_1)$ and $y \in L(R_2 + R_3) = L(R_2) \cup L(R_3)$.

We now simplify the RHS. Clearly, $w \in L(R_1R_2 + R_1R_3) = L(R_1R_2) \cup L(R_1R_3) = (L(R_1) \cdot L(R_2)) \cup (L(R_1) \cdot L(R_3)) \iff$ either $w = xy$ for some $x \in L(R_1)$ and $y \in L(R_2)$ or $w = xy$ for some $x \in L(R_1)$ and $y \in L(R_3) \iff w = xy$ for some $x \in L(R_1)$ and $y \in L(R_2) \cup L(R_3)$.

Hence, we conclude from the above two that $x \in L(R_1(R_2 + R_3)) \iff x \in L(R_1R_2 + R_1R_3)$. Hence, $R_1(R_2 + R_3) = R_1R_2 + R_1R_3$.

(b) $(R_1 + R_2)R_3 = R_1R_3 + R_2R_3$.

Solution: To show that the two regular expressions are equal, we show that the languages defined by the two are equal.

$$
\begin{aligned}
x \in L((R_1 + R_2)R_3) &\iff x \in L(R_1 + R_2) \cdot L(R_3) \\
&\iff \exists y \exists z, x = yz, y \in L(R_1 + R_2) \wedge z \in L(R_3) \\
&\iff \exists y \exists z, x = yz, (y \in L(R_1) \vee y \in L(R_2)) \wedge z \in L(R_3) \\
&\iff \exists y \exists z, x = yz, \\
&\qquad (y \in L(R_1) \wedge z \in L(R_3)) \vee (y \in L(R_2) \wedge z \in L(R_3)) \\
&\iff x \in L(R_1 R_3) \vee x \in L(R_2 R_3) \\
&\iff x \in L(R_1 R_3 + R_2 R_3)
\end{aligned}
$$

(c) $(R_1 + R_2)^* = (R_1^* R_2^*)^*$.

Solution: We will try to show containment in both directions.

($\Rightarrow$) Suppose that $w \in L((R_1 + R_2)^*)$. Then we can write $w = w_1 \cdot w_2 \cdots w_k$ for some $k \in \mathbb{N}$, and each $w_i \in L(R_1) \cup L(R_2)$. Thus, each $w_i \in L(R_1)^* L(R_2)^*$ as well - this is because if $w_i \in L(R_1)$, then $w\varepsilon \in L(R_1) \cdot L(R_2)^*$. Hence $w_i \in L(R_1^* R_2^*)$. Thus, $w \in L((R_1^* R_2^*)^*)$.

($\Leftarrow$) Let $w \in L((R_1^* R_2^*)^*)$. Then, $w = w_1 \cdot w_2 \cdots w_k$, where $w_i \in L(R_1^* R_2^*)$. Each $w_i = w_i^1 \cdot w_i^2 \cdots w_i^\ell$ where $w_i^j \in L(R_1 + R_2)$. Thus $w = L((R_1 + R_2)^*)^*$, and hence $w \in L((R_1 + R_2)^*)$.

A regex R is said to be (+)free if R does not contain the + operator. For instance $(0^* 1^*)^*$ is (+)-free. A regex R is said to be (+)-separable if one of the following two conditions hold.

- R is (+)-free, or
- $R = R_1 + R_2$, and R_1, R_2 are (+)-separable

Use Parts (a), (b), and (c) to prove the following questions.

(d) Show that if R is (+)-separable, then $R = R_1 + R_2 + \cdots + R_k$ where each of the R_is are (+)-free.

Solution: We use induction on size of R. Here size can be thought of number of + terms in R.

- **Base Case:** R is (+)-free. This is trivial with $k = 1$.
- **Inductive Hypothesis:** All R's with size $\leq n$ can be written as $R = R_1 + R_2 + \cdots + R_k$ where each of the R_is are (+)-free for some k.
- **Inductive Step:** Consider R of size $n + 1$. Now $R = R_1 + R_2$, where R_1,R_2 are (+)-separable (by definition). Applying Inductive Hypothesis on R_1 and R_2, we get $R_1 = X_1 + X_2 + \cdots + X_{k_1}$ and $R_2 = Y_1 + \cdots + Y_{k_2}$. This means $R = X_1 + X_2 + \cdots + X_{k_1} + Y_1 + \cdots + Y_{k_2}$ where each X_i and Y_j are (+)-free. Hence proved.

(e) Show that if R_1 and R_2 are (+)-separable, then there is a (+)-separable regex equivalent to R_1R_2.

Solution: Parts (a) and (b) can be extended for any $k \in \mathbb{N}$ to obtain:

$$R_1(R_2 + R_3 + \cdots + R_{k+1}) = R_1R_2 + R_1R_3 + \cdots R_1R_{k+1}$$
$$(R_1 + R_2 + \cdots + R_k)R_{k+1} = R_1R_{k+1} + R_2R_{k+1} + \cdots + R_kR_{k+1}$$

If R_1, R_2 are (+)-free, notice that so is R_1R_2.

Now, given (+)-separable R_1, R_2, from part (d), we have

$$R_1 = R_{1,1} + R_{1,2} + \cdots + R_{1,k_1}$$
$$R_2 = R_{2,1} + R_{2,2} + \cdots + R_{2,k_2}$$

for some $k_1, k_2 \in \mathbb{N}$, where all the regexes on the RHS are (+)-free. Then,

$$\begin{aligned} R_1R_2 &= R_1\left(\mathop{\Large +}_{j=1}^{k_2} R_{2,j}\right) \\ &= \mathop{\Large +}_{j=1}^{k_2} R_1R_{2,j} \\ &= \mathop{\Large +}_{j=1}^{k_2}\left(\mathop{\Large +}_{i=1}^{k_1} R_{1,i}\right)R_{2,j} \\ &= \mathop{\Large +}_{j=1}^{k_2}\mathop{\Large +}_{i=1}^{k_1} R_{1,i}R_{2,j} \\ &= \mathop{\Large +}_{i=1}^{k_1}\mathop{\Large +}_{j=1}^{k_2} R_{ij} \end{aligned}$$

where $R_{ij} = R_{1,i}R_{2,j}$ is (+)-free. Hence, R_1R_2 is (+)-free.

Here, $\mathop{+}_{i=1}^{n} R_i = R_1 + R_2 + \cdots + R_n$ has been used to succintly represent a serial + operation on the regexes $R_1, \cdots, R_n$.

(f) Show that if R is (+)-separable, then there is a (+)-free regex R_1 such that $L(R^*) = L(R_1^*)$.

Solution: The following generalization of part (c) can be easily shown via a similar method to part (c) -

$$(R_1 + R_2 + \cdots + R_k)^* = (R_1^*R_2^* \ldots R_k^*)^*$$

Now let R is a (+)-separable regex. Then $R = R_1 + R_2 + \cdots + R_k$ where each R_is are (+)-free. Hence, we have using the above result,

$$R^* = (R_1 + R_2 + \cdots + R_k)^* = (R_1^*R_2^* \ldots R_k^*)^* = R'^*$$

where $R' = R_1^* R_2^* \dots R_k^*$. Clearly, since each R_i is (+)-free, each R_i^* is (+)-free and hence R' is (+)-free. Hence, we have $L(R^*) = L(R'^*)$ where R' is (+)-free.