Regular Expressions

Definition 1.11 Let Σ be an alphabet. A **regular expression** (RE) R over the alphabet Σ describes a language $L(R)$ over Σ. It is inductively defined such that R is RE if R is

1. a for some $a \in \Sigma$, with $L(a) = \{a\}$
2. ε, with $L(\varepsilon) = \{\varepsilon\}$
3. $\emptyset$, with $L(\emptyset) = \emptyset$
4. $(R_1 \cup R_2)$ (R_1, R_2 REs), with $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$
5. $(R_1 \circ R_2)$ (R_1, R_2 REs), with $L(R_1 \circ R_2) = L(R_1) \circ L(R_2)$
6. (R_1^*), (R_1 RE), with $L(R_1^*) = L(R_1)^*$

Parentheses in an RE can be omitted. Then, evaluation is done in the precedence order $*$, $\circ$, $\cup$. If clear from the context, the concatenation operator $\circ$ does not have to be written down. Moreover, "$R_1 = R_2$" means $L(R_1) = L(R_2)$.

$$(0 \cup 1)0^*$$

$$L(0^*10^*) = \{w \in \Sigma^* | w \text{ has exactly a single } 1\}.$$

Theorem 1.5 A language is regular if and only if some regular expression describes it.

$$L = \{0^n 1^n \mid n \geq 0\}$$

1 Reading and understanding regular expressions

Let the alphabet $\Sigma = \{0,1\}$ be given. We consider the regular expressions

- $R_1 = \Sigma\Sigma^*$, and
- $R_2 = 0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$.

1. Are the strings 101, 100, ε contained in the languages described by these regular expressions?
2. Give the languages described by these regular expressions.

1. 1) R_1: $101 \in L(R_1)$ ✓ $\quad 100 \in L(R_1)$ ✓ $\quad \varepsilon \notin L(R_1)$

2) R_2 $\quad 101 \in L(R_2)$ ✓ $\quad 100 \notin L(R_2)$ ✓ $\quad \varepsilon \notin L(R_2)$ $\quad 0 \in L(R_2)$?

2. 1) $L(R_1) = \{ w \in \Sigma^* \mid w$ string of length at least $1\}$

$L(R_2) = \{ w \in \Sigma^n, n \geq 1 \mid w$ starts and ends with the same symbol$\}$

$L(\emptyset \circ R) = \emptyset$ $\qquad L_1 \circ L_2 = \{ w_1 w_2 \mid w_1 \in L_1$ and $w_2 \in L_2\}$

$L(\varepsilon \circ R)$

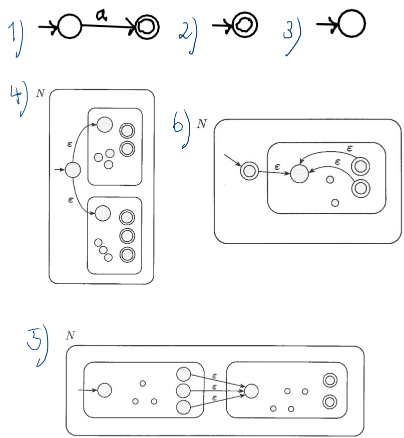
Lemma 1.1 If a language is described by a regular expression, then it is regular.

Definition 1.11 Let Σ be an alphabet. A **regular expression** (RE) R over the alphabet Σ describes a language $L(R)$ over Σ. It is inductively defined such that R is RE if R is

1. a for some $a \in \Sigma$, with $L(a) = \{a\}$
2. ε, with $L(\varepsilon) = \{\varepsilon\}$
3. $\emptyset$, with $L(\emptyset) = \emptyset$
4. $(R_1 \cup R_2)$ (R_1, R_2 REs), with $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$
5. $(R_1 \circ R_2)$ (R_1, R_2 REs), with $L(R_1 \circ R_2) = L(R_1) \circ L(R_2)$
6. (R_1^*), (R_1 RE), with $L(R_1^*) = L(R_1)^*$

Parentheses in an RE can be omitted. Then, evaluation is done in the precedence order $*$, $\circ$, $\cup$. If clear from the context, the concatenation operator $\circ$ does not have to be written down. Moreover, "$R_1 = R_2$" means $L(R_1) = L(R_2)$.

Proof idea

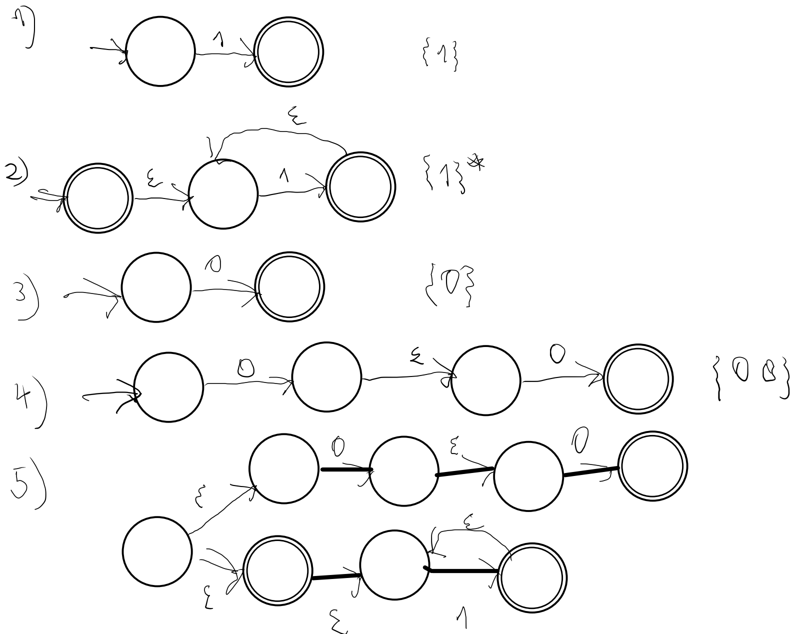


2 From RE to NFA

You are given the following regular expression over $\Sigma = \{0, 1\}$.

$$R = 1^* \cup 00$$

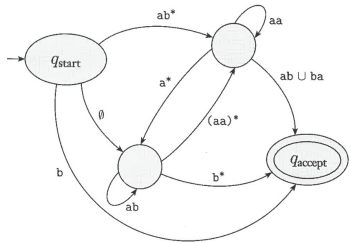
Use exactly the construction of the proof for Lemma 1.1 to build an NFA that recognizes $L(R)$.



Lemma 1.2 If a language is regular, then it is decribed by a regular expression.

Definition 1.12 A **generalized nondeterministic finite automaton** (GNFA), $(Q, \Sigma, \delta, q_{start}, q_{accept})$, is a 5-tuple, with

1. Q is the finite set of states,
2. Σ is the finite input alphabet,
3. $\delta : Q \times Q \to \mathcal{R}$ ($\mathcal{R}$: set of all REs over Σ), where $\delta(q, q_{start})$ and $\delta(q_{accept}, q)$ are undefined for all $q \in Q$.
4. $q_{start} \in Q \setminus \{q_{accept}\}$ is the start state, and
5. $q_{accept} \in Q \setminus \{q_{start}\}$ is the accept state.



Definition 1.13 A GNFA $G = (Q, \Sigma, \delta, q_{start}, q_{accept})$ **accepts** a string $w \in \Sigma^*$, if there exists a decomposition $w = w_1 w_2 \cdots w_k$ ($w_i \in \Sigma^*$) and a sequence of states $q_0, q_1, \ldots, q_k$ such that

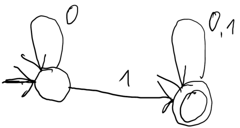
1. $q_0 = q_{start}$ is the start state
2. $q_k = q_{accept}$ is the accept state, and
3. for each $i = 1, \ldots, k$ we have $w_i \in L(R_i)$, where $R_i = \delta(q_{i-1}, q_i)$.

Otherwise w is **rejected**. $L(G)$ is the **language recognized by** G.

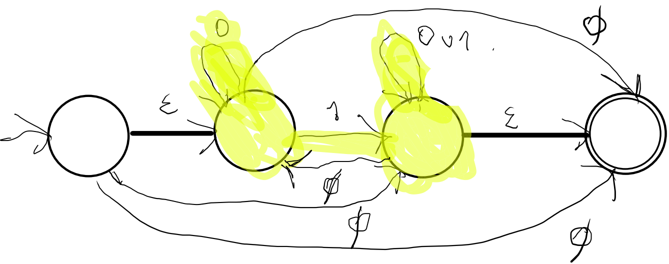
Lemma 1.3 For each FA M, there is a GNFA G, such that $L(G) = L(M)$.

3 From FA to RE

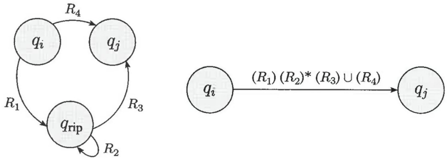
Let the FA M be given by its STD:



1. Convert M to a GNFA G with $L(G) = L(M)$ using exactly the construction in the proof of Lemma 1.3
2. Convert G to a RE R with $L(R) = L(G)$ using exactly the construction in the proof of Lemma 1.4.



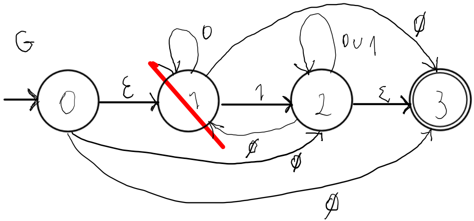
Lemma 1.4 For each GNFA G, there is a regular expression R, such $L(R) = L(G)$.

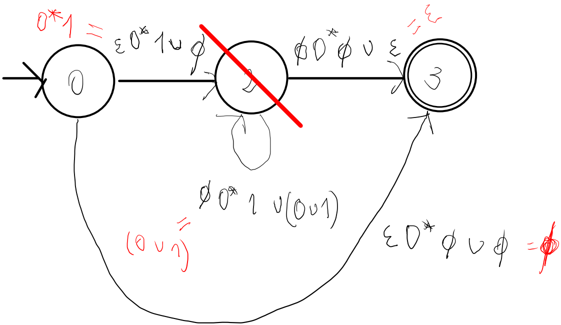


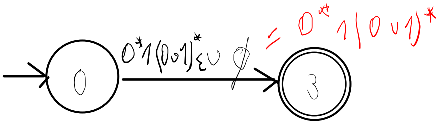
function CONVERT($G = (Q, \Sigma, \delta, q_{start}, q_{accept})$)
    $k \leftarrow |Q|$
    if $k = 2$ **then**
        return $\delta(q_{start}, q_{accept})$
    else
        if $k > 2$ **then**:
            select $q_{rip} \in Q \setminus \{q_{start}, q_{accept}\}$
            $Q' \leftarrow Q \setminus \{q_{rip}\}$
            for all $q_i \in Q' \setminus \{q_{accept}\}, q_j \in Q' \setminus \{q_{start}\}$ **do**
                $R_1 \leftarrow \delta(q_i, q_{rip})$
                $R_2 \leftarrow \delta(q_{rip}, q_{rip})$
                $R_3 \leftarrow \delta(q_{rip}, q_j)$
                $R_4 \leftarrow \delta(q_i, q_j)$
                $\delta'(q_i, q_j) \leftarrow (R_1)(R_2)^*(R_3) \cup (R_4)$
            end for
            $G' \leftarrow (Q', \Sigma, \delta', q_{start}, q_{accept})$
            return CONVERT(G')
        end if
    end if
end function

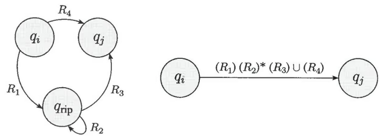
3 From FA to RE (cont.)

2. Convert G to a RE R with $L(R) = L(G)$ using exactly the construction in the proof of Lemma 1.4.









$\{0^n 1^n \mid n \geq 1\} = L$ not regular

$L(00^* 11^*) \neq$

0111111

$(a \circ a a)^* \cup \ldots$