Continuation on formal languages

Any string s can be considered as a concatenation of three substrings:

$$s = tuv \quad \text{where}$$

t = prefix of s
u = substring of s
v = suffix of s

Ex $s = abc$

Examples of prefixes

$t = \varepsilon, a, ab, abc$

suffixes $v = \varepsilon, c, bc, abc$

substrings $u = \varepsilon, a, b, c, ab, bc, abc$

Extended transition function

$$\delta(q, \varepsilon) = q$$

$$\delta(q, sv) = \delta(\delta(q, s), v)$$

$q_1 = \delta(q_1, \varepsilon) \qquad q_2 = \delta(q_1, a)$

$q_3 = \delta(q_2, b) = \delta(\delta(q_1, a), b) = \delta(q_1, ab)$

$q_4 = \delta(q_1, abc)$

Operations on languages

concatenation:

$L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Sigma^*$

$L_1 L_2 = \{ s \in \Sigma^* \mid s = s_1 s_2 \wedge s_1 \in L_1 \wedge s_2 \in L_2 \}$

Ex $L_1 = \{ s_1, t_1 \}$ $L_2 = \{ s_2, t_2 \}$

$L_1 L_2 = \{ s_1 s_2, s_1 t_2, t_1 s_2, t_1 t_2 \}$

Kleene closure:

$L^* = \{ \varepsilon \} \cup L \cup LL \cup LLL \cup \dots$

$= \bigcup_{k=0}^{N} L^k$ N is arbitrary large but not equal to infinity

Ex $L = \{ abc \}$

$L^* = \{ \varepsilon, abc, abcabc, \dots \} = (abc)^*$

Prefix closure

$\bar{L} = \{ s \in \Sigma^* \mid \exists t \in \Sigma^* \wedge st \in L \}$

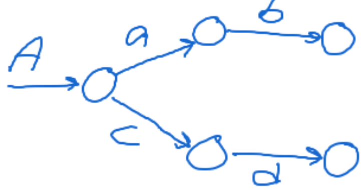
Ex $L = \{ abc \}$

$\bar{L} = \{ \varepsilon, a, ab, abc \}$

Ex $K = \{ ab \}$ $L = \{ ab, cd \}$

$\bar{K} = \{ \varepsilon, a, ab \}$ $\bar{L} = \{ \varepsilon, a, ab, c, cd \}$

$\bar{K} = \mathcal{L}(S)$ $\bar{L} = \mathcal{L}(A)$



$\bar{K}\Sigma_u \cap L \subseteq \bar{K}$?

1) $\Sigma_u = \{b\}$

$\underbrace{\{\varepsilon, a, ab\}\{b\}}_{\{b, ab, abb\}} \cap \{\varepsilon, a, ab, c, cd\} =$

$= \{ab\} \subseteq \{\varepsilon, a, ab\} = \bar{K}$

2) $\Sigma_u = \{a, c\}$

$\{\varepsilon, a, ab\}\{a, c\} \cap \{\varepsilon, a, ab, c, cd\}$

$= \dots = \{a, c\} \not\subseteq \bar{K} = \{\varepsilon, a, ab\}$

Regular languages

Every automaton can be represented by a regular language and the opposite.

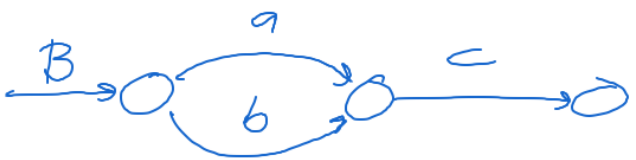
A regular language is defined by the following three operators:

1) concatenation of string s_1 followed by string s_2: $s_1 s_2$
2) union of two strings s_1 and s_2: $s_1 + s_2 = \{s_1\} \cup \{s_2\}$
3) closure of string s: $s^* = \{\varepsilon, s, ss, sss, \dots\}$
4) closure of string s but at least one s: $s^+ = \{s, ss, sss, \dots\}$

The forth operator s^+ is a derived operator since $ss^* = s^+$

Ex

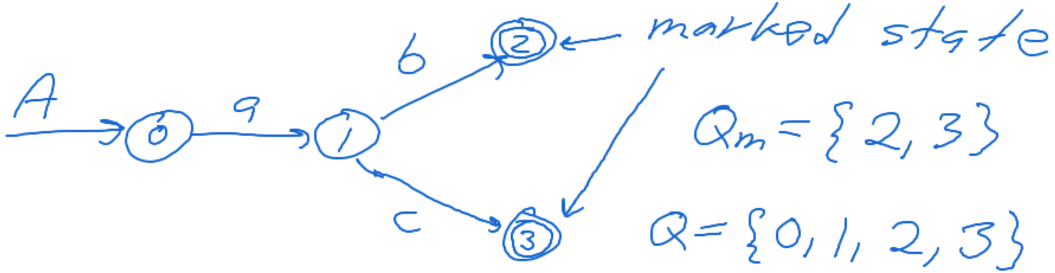
$L(A) = \overline{abc} = \{\varepsilon, a, ab, abc\}$



$L(B) = \overline{ac + bc} = \overline{(a+b)c}$

$L(C) = \overline{(abb^*c)^*} = \overline{(ab^+c)^*}$

Marked states and marked languages



$Q_m = \{2, 3\}$

$Q = \{0, 1, 2, 3\}$

$A = \langle Q, \Sigma, \delta, q_i, Q_m \rangle$

Q_m = set of marked states

If no marked state are defined $\Rightarrow Q_m = Q$, i.e. all states are marked.

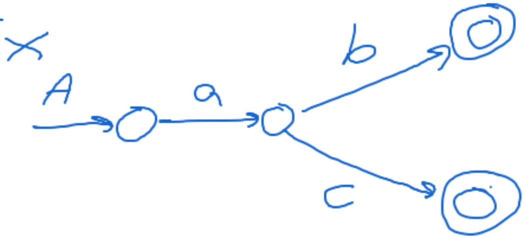
Marked language

Def. $L_m(A) = \{s \in L(A) \mid \delta(q_i, s) \in Q_m\}$

Prefix closed language

$$L(A) = \overline{L(A)}$$

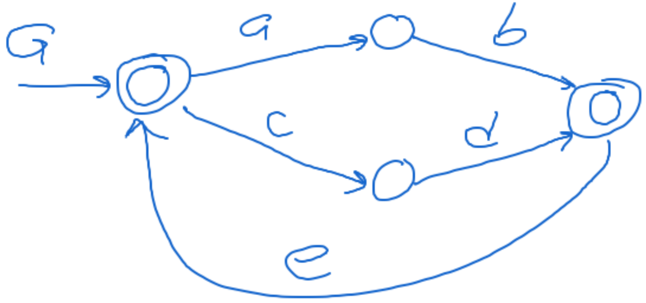
Ex



$L_m(A) = a(b+c) = ab + ac$

$L(A) = \{\varepsilon, a, ab, ac\} = \overline{a(b+c)}$

$= \overline{L_m(A)}$



$L_m = \{\varepsilon, \underbrace{ab+cd}_{s}, se, ses, sese, \dots\}$

$= (se)^*(\varepsilon + s)$

$L(G) = \overline{(se)^*(\varepsilon + s)}$

Ex

This is an example of a blocking system we want to avoid.

$L(A) = \overline{abb^*} = \overline{ab^+}$

$L_m(A) = a \quad \therefore \overline{L_m(A)} \neq L(A)$