Closure properties of regular languages

Let L and M be regular languages over Σ. Then the following languages are all regular

- Union: $L \cup M$
- Intersection: $L \cap M$
- Complement: $\overline{L} = \Sigma^* \setminus L$
- Difference: $L \setminus M$
- Reversal: $L^R = \{w^R \mid w \in L\}$
- Kleene closure: L^*
- Concatenation: $L.M$
- Homomorphism: $h(L) = \{h(w) \mid w \in L\}$
- Inverse homomorphism: $h^{-1}(L) = \{w \in \Sigma^* \mid h(w) \in L\}$

Closure under union

Theorem For any regular languages L e M, $L \cup M$ is regular

Proof Let E and F be regular expressions such that $L = L(E)$ and $M = L(F)$. Then $L \cup M$ is generated by $E + F$, and is regular by definition $\square$

Closure under concatenation and Kleene

The proof of closure under union is rather **immediate**, since regular expressions use the union operator

Similarly, we can immediately prove the closure under

- concatenation
- Kleene operator

Closure under complement

Theorem If L is a regular language over Σ, then so is $\overline{L} = \Sigma^* \setminus L$

Proof Let L be recognized by a DFA

$$A = (Q, \Sigma, \delta, q_0, F).$$

Let $B = (Q, \Sigma, \delta, q_0, Q \setminus F)$. Now $L(B) = \overline{L}$ $\square$

Closure under intersection

Theorem If L and M are regular, then so is $L \cap M$

Proof By De Morgan's law, $L \cap M = \overline{\overline{L} \cup \overline{M}}$

We already know that regular languages are closed under complement and union $\square$

Intersection automaton

Proof (alternative) Let $L = L(A_L)$ and $M = L(A_M)$ for automata A_L and A_M with

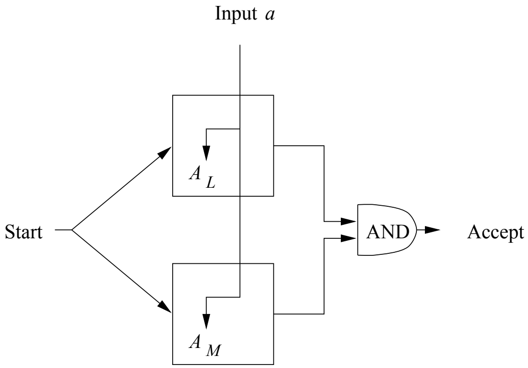
$$A_L = (Q_L, \Sigma, \delta_L, q_L, F_L)$$
$$A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$$

Without any loss of generality, we assume that both automata are deterministic

We shall construct an automaton that simulates A_L and A_M in parallel, and accepts if and only if both A_L and A_M accept

Intersection automaton

Idea : If A_L goes from state p to state s upon reading a, and A_M goes from state q to state t upon reading a, then $A_{L\cap M}$ will go from state (p, q) to state (s, t) upon reading a



Intersection automaton

Formally

$$A_{L\cap M} = (Q_L \times Q_M, \Sigma, \delta_{L\cap M}, (q_{L,0}, q_{M,0}), F_L \times F_M),$$

where

$$\delta_{L\cap M}((p,q),a) = (\delta_L(p,a), \delta_M(q,a))$$

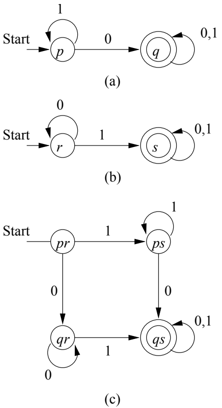
We can show by induction on $|w|$ that

$$\hat{\delta}_{L\cap M}((q_{L,0}, q_{M,0}), w) = \left(\hat{\delta}_L(q_{L,0}, w), \hat{\delta}_M(q_{M,0}, w)\right)$$

Then $A_{L\cap M}$ accepts if and only if A_L and A_M accept $\square$

Exercise

Build an automaton that accepts strings with at least one 0 and at least one 1. Let's build **simpler** automata and take the intersection



Closure under set difference

Theorem If L and M are regular languages, so is $L \setminus M$

Proof Observe that $L \setminus M = L \cap \overline{M}$

We already know that regular languages are closed under complement and intersection □

Closure under reverse operator

Theorem If L is regular, so is L^R

Proof Let L be recognized by FA A. Turn A into an FA for L^R by

- reversing all arcs
- make the old start state the new sole accepting state
- create a new start state p_0 such that $\delta(p_0, \epsilon) = F$, F the set of accepting states of old A □

Closure under reverse operator

Proof (alternative) Let E be a regular expression. We shall construct a regular expression E^R such that $L(E^R) = (L(E))^R$

We proceed by structural induction on E

Base If E is ϵ, $\emptyset$, or $\boldsymbol{a}$, then $E^R = E$ (easy to verify)

Closure under reverse operator

Induction

- $E = F + G$: We need to reverse the two languages. Then $E^R = F^R + G^R$
- $E = F.G$: We need to reverse the two languages and also reverse the order of their concatenation. Then $E^R = G^R.F^R$
- $E = F^*$:
  $w \in L(F^*)$ means $\exists k \; : \; w = w_1 w_2 \cdots w_k, \; w_i \in L(F)$
  then $w^R = w_k^R w_{k-1}^R \cdots w_1^R, \; w_i^R \in L(F^R)$
  then $w^R \in L(F^R)^*$
  Same reasoning for the inverse direction. Then $E^R = (F^R)^*$

Thus $L(E^R) = (L(E))^R$ $\square$