Automata and Formal Languages

The Regular Operations (continued)

- Proof of closure under union:

Theorem: Regular languages are closed under union.

Proof: We prove the theorem by construction

Assuming the alphabet of M_1 is the same as M_2 for simplicity but the proof can be extended for different Σ.

$M_1 = \{Q_1, \Sigma, \delta_1, q_1, F_1)$

$M_2 = \{Q_2, \Sigma, \delta_2, q_2, F_2)$

$M = \{Q, \Sigma, \delta, q_0, F)$

We construct $L(M) = A_1 \cup A_2$

$Q = Q_1 \times Q_2$

$q_0 = (q_1, q_2)$

$F = (Q_1 \times F_2) \cup (F_1 \times Q_2) = \{(r, s) \,|\, r \in F_1 \cup s \in F_2\}$

$\sigma \in \Sigma$

$(q_1, q_2) \in Q$

$\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$

Definition of the complement operator:

$\overline{A} = \{x \,|\, x \notin A\}$

$A \cap B = \{x \,|\, x \in A \cap x \in B\}$

Nondeterminism

In a **DFA** for every input the machine has only one possible move:

—> Given an input string, the sequence of visited states is *unique*.

In a **NFA** the machine can follow multiple paths in parallel.

When we reach a state where multiple ways to proceed exists:

The machine splits into multiple copies of itself and follows all the possibilities in parallel.

Each copy goes on as before.

If the next symbol is not any of the arrows exiting a state where a copy is: the copy dies.

If any of the copies ens up in an accept state at the end of the input, the NFA accepts the input string.

When we reach a state where an empty string ϵ transitions exists, the machine splits again (one stays, one follows the transition)