Automata & languages

A primer on the Theory of Computation

Laurent Vanbever
nsg.ee.ethz.ch

ETH Zürich (D-ITET)
30 September 2021
Part 2 out of 4
Last week was all about

Deterministic Finite Automaton
We saw three main concepts

Regular Language

Formal definition

Closure
A language $L$ is regular if some finite automaton recognizes it.
A finite automaton is a 5-tuple

\((Q, \Sigma, \delta, q_0, F)\)
A minimal deterministic finite automaton is defined as:

\[(Q, \Sigma, \delta, q_0, F)\]

where
- \(Q\) is the set of states,
- \(\Sigma\) is the alphabet,
- \(\delta\) is the transition function,
- \(q_0\) is the start state,
- \(F\) is the set of accept states.
Regular Language

Formal definition

Closure

If \( L_1 \) and \( L_2 \) are regular, then so are:

\[
L_1 \cup L_2 \quad L_1 \cap L_2 \quad \overline{L_1} \\
L_1 \oplus L_2 \quad L_1 - L_2
\]
Finite Automata
Thu Sept 30

1 Closure

2 Equivalence
   - DFA
   - NFA
   - Regular Expression