Regular Languages

CS60001: Foundations of Computing Science

Pallab Dasgupta
Professor, Dept. of Computer Sc. & Engg.,
Indian Institute of Technology Kharagpur
A deterministic finite automaton (DFA) is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\), where

- \(Q\) is a finite set called the states,
- \(\Sigma\) is a finite set called the alphabet,
- \(\delta: Q \times \Sigma \rightarrow Q\) is the transition function,
- \(q_0 \in Q\) is the start state, and
- \(F \subseteq Q\) is the set of accepted states (final states).

Example: \(M = (Q, \Sigma, \delta, q_1, F)\), where

- \(Q = \{q_1, q_2, q_3\}\),
- \(\Sigma = \{0,1\}\),
- \(\delta\) is described as
- \(q_1\) is the start state
- \(F = \{q_2\}\)
Acceptance/Recognition by DFA

- Let $M = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton and $w = w_1w_2...w_n$ be a string where each $w_i \in \Sigma$. Then $M$ accepts $w$ if a sequence of states $r_0, r_1, ..., r_n$ in $Q$ exists with three conditions:
  - $r_0 = q_0$,
  - $\delta(r_i, w_{i+1}) = r_{i+1}$, for $i = 0, 1, ..., n-1$, and
  - $r_n \in F$

Therefore, $M$ recognizes language $A_M$ if $A_M = \{w \mid M$ accepts $w\}$

- Example:

$L(M_1) = A_{M1}$ ($M_1$ recognizes/accepts $A_{M1}$), where $A_{M1} = \{w \mid w$ contains at least one 1 and an even number of 0s follow the last 1$\}$
A non-deterministic finite automaton (NFA) is a 5-tuple \((Q, \sum, \delta, q_0, F)\), where

- \(Q\) is a finite set called the states,
- \(\sum\) is a finite set called the alphabet,
- \(\delta: Q \times \sum \rightarrow P(Q)\) is the transition function,
- \(q_0 \in Q\) is the start state, and
- \(F \subseteq Q\) is the set of accepted states (final states).

Example: \(N = (Q, \sum, \delta, q_1, F)\), where

- \(Q = \{q_1, q_2, q_3, q_4\}\),
- \(\sum = \{0,1\}\),
- \(\delta\) is described as
- \(q_1\) is the start state
- \(F = \{q_4\}\)
Acceptance/Recognition by NFA

Let \( N = (Q, \sum, \delta, q_0, F) \) be a non-deterministic finite automaton and \( y = y_1y_2...y_n \) be a string where each \( y_i \in \sum^\epsilon \). Then \( N \) accepts \( y \) if a sequence of states \( r_0, r_1, ..., r_m \) in \( Q \) exists with three conditions:

- \( r_0 = q_0 \),
- \( r_{i+1} \in \delta(r_i, y_{i+1}) \), for \( i = 0, 1, ..., m-1 \), and
- \( r_m \in F \)

Therefore, \( N \) recognizes language \( A_N \) if \( A_N = \{y \mid N \text{ accepts } y\} \)

Example:

\( L(N_1) = A_{N_1} \) (\( N_1 \) recognizes/accepts \( A_{N_1} \)), where
\( A_{N_1} = \{y \mid y \text{ contains either 101 or 11 as a substring}\} \)
A language is called a regular language if some automaton recognizes it.

Let A and B be regular languages. The regular operations union, concatenation and star are defined as follows:

- **Union:** \( A \cup B = \{ x \mid x \in A \text{ or } x \in B \} \)
- **Concatenation:** \( A \circ B = \{ xy \mid x \in A \text{ and } y \in B \} \)
- **Star:** \( A^* = \{ x_1x_2...x_k \mid k \geq 0 \text{ and } x_i \in A \} \)
Closure under Regular Operations

- **Closure Theorems:**
  - The class of regular languages is closed under the union operation
    (if \( A_1 \) and \( A_2 \) are regular languages, so is \( A_1 \cup A_2 \))
  - The class of regular languages is closed under the concatenation operation
    (if \( A_1 \) and \( A_2 \) are regular languages, so is \( A_1 \circ A_2 \))
  - The class of regular languages is closed under the star operation
    (if \( A \) is a regular language, so is \( A^* \))
Regular Expressions

- **R is a regular expression if R is**
  - a for some a in the alphabet ∑,
  - ε,
  - Φ,
  - (R₁ U R₂), where R₁ and R₂ are regular expressions
  - (R₁ ○ R₂), where R₁ and R₂ are regular expressions
  - R₁*, where R₁ is a regular expression

- **Some Important Identities:**
  - R⁺ ≡ RR* and R⁺ U ε ≡ R*
  - R U Φ ≡ R and R ○ ε ≡ R
  - (R U ε) may not equal R (Ex: if R = 0; then L(R) = {0}, but L(R U ε) = {0, ε})
  - (R ○ Φ) may not equal R (Ex: if R = 0; then L(R) = {0}, but L(R ○ Φ) = Φ)

- **Example of Regular Expression**
  - Let D = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} is the alphabet of decimal digits; then a numerical constant that may include a fractional part and/or a sign may be described as a member of the language: (+ U – U ε) (D⁺ U D⁺ . D* U D* . D+)
Two finite automata are *equivalent* if they accept the same regular language.

**Theorems:**
- Every non-deterministic finite automaton has an equivalent deterministic finite automaton.
- A language is regular if and only if some non-deterministic finite automaton recognizes/accepts it.
- A language is regular if and only if some regular expression describes it.
- If a language $L$ is accepted by a DFA, then $L$ is denoted by a regular expression.
Let \( r \) be a regular expression. Then there exists an NFA with \( \varepsilon \)-transitions (\( M \)) that accepts \( L(r) \). The construction procedure is as follows:

- For Union: \( L(M) = L(M_1) \cup L(M_2) \)
- For Concatenation: \( L(M) = L(M_1) \circ L(M_2) \)
- For Star: \( L(M) = L(M_1)^* \)
Pumping Lemma: Proving Non-regularity

- If \( A \) is a regular language, then there is a number \( p \) (the pumping length) where, 
  - \( s \) is any string in \( A \) of length at least \( p \), then \( s \) may be divided into three pieces, 
  - \( s = xyz \) satisfying the following conditions:
    - For each \( i \geq 0 \), \( xy^i z \in A \)
    - \( |y| > 0 \), and
    - \( |xy| \leq p \)

- Examples:
  - The following languages (denoted by \( B, C, D, E, F \)) are not regular:
    - \( B = \{0^n1^n | n \geq 0\} \)
    - \( C = \{w | w \text{ has an equal number of 0s and 1s}\} \)
    - \( F = \{ww | w \in \{0, 1\}^*\} \)
    - \( D = \{1^{n^2} | n \geq 0\} \)
    - \( E = \{0^i1^j | i > j\} \)