Review Regular Expressions

Method to represent strings in a language

+ union (or)
⊙ concatenation (AND) (can omit)
* star-closure (repeat 0 or more times)

Example:

\[(a + b)^* \circ a \circ (a + b)^* = (a + b)^*a(a + b)^*\]

Closure Properties

A set is closed over an operation if

\[L_1 \cap L_2 \in \text{class} \]
\[L_1 \circ L_2 = L_3 \]
\[\Rightarrow L_3 \in \text{class} \]

Example

\[L_1 = \{x \mid x \text{ is a positive even integer}\} \]

\[L\] is closed under

addition?
multiplication?
subtraction?
division?

Example

\[L_2 = \{x \mid x \text{ is a positive odd integer}\} \]

\[L\] is closed under

addition?
multiplication?
subtraction?
division?
Closure of Regular Languages

Theorem 2.3.1 If $L_1$ and $L_2$ are regular languages then

$$L_1 \cup L_2$$
$$L_1L_2$$
$$L_1^*$$
$$L_1'$$
$$L_1 \cap L_2$$

are regular languages.

Proof (sketch)

Union $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1) \Gamma M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$

Construct $M(M) = L(M_1) \cup L(M_2)$

Concatenation $M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1) \Gamma M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$

Construct $M(M) = L(M_1) \circ L(M_2)$

Kleene Star

$M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1)$

Construct $M(M) = L(M_1)^*$

Complementation:

$M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$

Construct $M(M) = L(\bar{M}_1)$

Intersection

$M_1 = (K_1, \Sigma, \Delta_1, s_1, F_1) \Gamma M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$

Construct $M(M) = L(M_1) \cap L(M_2)$
Example:

Regular languages are closed under
- reversal $L^R$
- difference $L_1 \setminus L_2$
- right quotient $L_1 / L_2$
- homomorphism $h(L)$

Right quotient
Def: $L_1 / L_2 = \{ x | xy \in L_1 \text{ for some } y \in L_2 \}$

Example:

$$L_1 = \{ a^* b^* \cup b^* a^* \}$$
$$L_2 = \{ b^n | n \text{ is even}, n > 0 \}$$
$$L_1 / L_2 =$$

Homomorphism
Def. Let $\Sigma, \Gamma$ be alphabets. A homomorphism is a function

$$h : \Sigma \rightarrow \Gamma^*$$

Example:

$$\Sigma = \{ a, b, c \}, \Gamma = \{ 0, 1 \}$$
$$h(a) = 11$$
$$h(b) = 00$$
$$h(c) = 0$$

$$h(bc) =$$
$$h(ab^*) =$$

Example using the homomorphism above.

$L = a^* bb \Gamma h(L) =$

Equivalence of DFA and R.E.
**Definition** A language $L$ is regular if it can be described by a regular expression.

**Theorem 2.3.3** A language is regular if and only if it is accepted by a finite automaton.

- **Proof Part 1 ($\Rightarrow$):**
  Let $r$ be a R.E. then $\exists$ NFA $M$ s.t. $L(M) = L(r)$.
  0
  \{ε\}
  \{a\}
  Suppose $r$ and $s$ are R.E.
  1. $r+s$
  2. $rs$
  3. $r^*$

**Example**

$a b^* + a$

- **Proof Part 2 ($\Leftarrow$):**
  Given an NFA $M \exists$ R.E. $r$ s.t. $L(M) = L(r)$.

**Example:**

![Diagram of a finite automaton](image)

**Grammar** $G=(V, \Sigma, R, S)$

- $V$ variables (nonterminals)
- $\Sigma$ terminals
- $R$ rules (productions)
- $S$ start symbol

**Right-linear grammar:**

all productions of form

$A \rightarrow xB$

$A \rightarrow x$

where $AxB \in V^*x \in \Sigma^*$
Left-linear grammar:

all productions of form
A → Bx
A → x
where A/B ∈ V/x ∈ Σ*

Definition:
A regular grammar is a right-linear or left-linear grammar.

Example 1:

G = (V{S, a, b}, Σ, {S}, {S → abS, S → ϵ, S → Sab})

Example 2:

G = (V{S, a, b}, Σ, {S}, {S → aB, B → bS | ϵ, B → aS | bB})

Theorem: L is a regular language iff ∃ regular grammar G s.t. L = L(G).

Outline of proof:

(⇐⇒) Given a regular grammar G
Construct NFA M
Show L(G) = L(M)
(⇒⇒) Given a regular language
∃ DFA M s.t. L = L(M)
Construct reg. grammar G
Show L(G) = L(M)

Proof of Theorem:

(⇐⇒) Given a regular grammar G
G = (V{S, a, b}, Σ, {S}, {S → abS, S → ϵ, S → Sab})

Assume G is right-linear
(left-linear case similar).
Construct NFA M s.t. L(G) = L(M)
If w ∈ L(G) then w = v₁v₂...vₖ
M=(VΣΔV₀Γ)
V₀ is the start (initial) state
For each production \( V_i \to aV_j \Gamma \)

For each production \( V_i \to a \Gamma \)

Show \( L(G)=L(M) \)
Thus given \( R, G, \Gamma \)
\( L(G) \) is regular

\[ (\implies) \] Given a regular language \( L \)
\( \exists \) DFA \( M \) s.t. \( L=L(M) \)
\[ M=(K\Sigma\DeltaK₀Γ) \]
\( K=\{q₀, q₁, \ldots, qₙ\} \)
\( \Sigma=\{a₁, a₂, \ldots, aₘ\} \)
Construct reg. grammar \( G \) s.t. \( L(G)=L(M) \)
\( G=(K\Sigma\DeltaRq₀) \)
if \( \delta(q_i, a_j)=q_k \) then

if \( q_k \in F \) then

Show \( w \in L(M) \iff w \in L(G) \)
Thus \( L(G)=L(M) \).
QED.

Example

\[ G=\{SIB|ÎR[aB]|bS|\lambda \}GR= \]
\( S \to aB | bS | \lambda \)
\( B \to aS | bB \)

Example: