Section: Regular Languages

Regular Expressions

Method to represent strings in a language

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

Example:

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

Example:

\[(aa)^*\]
Definition Given $\Sigma$, 

1. $\emptyset, \lambda, a \in \Sigma$ are R.E.

2. If $r$ and $s$ are R.E. then
   - $r+s$ is R.E.
   - $rs$ is R.E.
   - $r^*$ is R.E.

3. $r$ is a R.E. iff it can be derived from (1) with a finite number of applications of (2).
Definition: \( L(r) = \) language denoted by R.E. \( r \).

1. \( \emptyset, \{\lambda\}, \{a\} \) are \( L \) denoted by a R.E.

2. if \( r \) and \( s \) are R.E. then
   
   \( \begin{align*}
   (a) \quad L(r+s) &= L(r) \cup L(s) \\
   (b) \quad L(rs) &= L(r) \circ L(s) \\
   (c) \quad L((r)^*) &= (L(r)^*)
   \end{align*} \)
Precedence Rules

* highest

Example:

\[ ab^* + c = \]

Examples:

1. \( \Sigma = \{a, b\}, \{w \in \Sigma^* \mid w \text{ has an odd number of } a\text{'s followed by an even number of } b\text{'s}\} \).

2. \( \Sigma = \{a, b\}, \{w \in \Sigma^* \mid w \text{ has no more than 3 } a\text{'s and must end in } ab\} \).

3. Regular expression for positive and negative integers
Section 3.2 Equivalence of DFA and R.E.

Theorem Let \( r \) be a R.E. Then \( \exists \) NFA \( M \) s.t. \( L(M) = L(r) \).

- Proof:
  \[
  \emptyset \\
  \{\lambda\} \\
  \{a\}
  \]
  Suppose \( r \) and \( s \) are R.E.

1. \( r + s \)
2. \( r \circ s \)
3. \( r^* \)
Example

$ab^* + a$

Theorem Let $L$ be regular. Then $\exists$ R.E. $r$ s.t. $L=L(r)$.

- Proof:
  - $L$ is regular
Example:
Grammar $G=(V,T,S,P)$

- $V$: variables (nonterminals)
- $T$: terminals
- $S$: start symbol
- $P$: productions

Right-linear grammar:

all productions of form

- $A \rightarrow xB$
- $A \rightarrow x$

where $A, B \in V$, $x \in T^*$
Left-linear grammar:

all productions of form
A \rightarrow Bx
A \rightarrow x

where A,B \in V, x \in T^*

Definition:

A regular grammar is a right-linear or left-linear grammar.
Example 1:

\[ G = (\{S\}, \{a, b\}, S, P), \ P = \]
\[
S \rightarrow abS \\
S \rightarrow \lambda \\
S \rightarrow Sab \\
\]

Example 2:

\[ G = (\{S, B\}, \{a, b\}, S, P), \ P = \]
\[
S \rightarrow aB \mid bS \mid \lambda \\
B \rightarrow aS \mid bB \\
\]
Theorem: \( L \) is a regular language iff \( \exists \) regular grammar \( G \) s.t. \( L = L(G) \).

Outline of proof:

(\( \iff \)) Given a regular grammar \( G \\
Construct NFA \( M \)  \\
Show \( L(G) = L(M) \)

(\( \implies \)) Given a regular language  \\
\( \exists \) DFA \( M \) s.t. \( L = L(M) \)  \\
Construct reg. grammar \( G \)  \\
Show \( L(G) = L(M) \)
Proof of Theorem:

\[(\iff)\] Given a regular grammar \( G \)
\[G = (V, T, S, P)\]
\[V = \{V_0, V_1, \ldots, V_y\}\]
\[T = \{v_o, v_1, \ldots, v_z\}\]
\[S = V_0\]
Assume \( G \) is right-linear
(see book for left-linear case).
Construct NFA \( M \) s.t. \( L(G) = L(M) \)
If \( w \in L(G) \), \( w = v_1v_2 \ldots v_k \)
\[ M = (V, T, \delta, V_0, F) \]
\[ V_0 \text{ is the start (initial) state} \]
For each production, \( V_i \rightarrow aV_j \),

For each production, \( V_i \rightarrow a \),

Show \( L(G) = L(M) \)

Thus, given R.G. G, 
\( L(G) \) is regular
(\implies\implies) \text{Given a regular language } L \\
\exists \text{ DFA } M \text{ s.t. } L=L(M) \\
M=(Q,\Sigma,\delta,q_0, F) \\
Q=\{q_0, q_1, \ldots, q_n\} \\
\Sigma = \{a_1, a_2, \ldots, a_m\} \\
\text{Construct R.G. } G \text{ s.t. } L(G) = L(M) \\
G=(Q,\Sigma,q_0,P) \\
\text{if } \delta(q_i,a_j)=q_k \text{ then} \\
\text{if } q_k \in F \text{ then} \\
\text{Show } w \in L(M) \iff w \in L(G) \\
\text{Thus, } L(G) = L(M). \\
QED.
Example

\[ G = (\{S, B\}, \{a, b\}, S, P), \quad P = \]
\[ S \rightarrow aB \mid bS \mid \lambda \]
\[ B \rightarrow aS \mid bB \]

Example: