Regular Set

Regular Expressions. - Syntax to define regular sets.

Finite state Automata - Computational model for processing regular sets.

Syntax specification for a PL.

<if-stmt> ::= if <bool-expr> then <statement> else <statement>

<arith-exp> ::= <var> | <const> |

<arith-exp> <arith-op><arith-exp>.

<arith-op> ::= + | - | * | /

<const> ::= 0 | 1 | 2 | -- | 8 | 9

<var> ::= a | b | c ... | x | y | z

$A = \{a^n b^n \mid n \geq 0\}$. - not regular.

$\downarrow$

a = [b =]

$A = \{a^n b^n \mid n \geq 0\}$. - not regular.

Recursive definition

- $\epsilon \in A$
- if $w \in A$ then $awb \in A$.

Context Free Languages.

Context Free Grammars.

$$G = (N, \Sigma, P, S).$$

N - finite set (non-terminal symbols).

Σ - finite set (terminal symbols). $\Sigma \cap N = \phi$

P - finite subset of $N \times (N \cup \Sigma)^*$ [productions]
$A \rightarrow \alpha$

$S \in N$ (Start Symbol).

Notation. A, B, C - non-terminals.
a, b, c - terminal symbols
α, β, γ - strings over $(N \cup \Sigma)^*$

Context Free Grammars.

$$G = (N, \Sigma, P, S).$$

N - finite set (non-terminal symbols).

Σ - finite set (terminal symbols). $\Sigma \cap N = \phi$

P - finite subset of $N \times (N \cup \Sigma)^*$ [productions]

$P \subseteq N \times (N \cup \Sigma)^*$ $\quad \{(A, \alpha_1), (A, \alpha_2), (A, \alpha_3)\} \subseteq P$

$A \to \alpha_1 \mid \alpha_2 \mid \alpha_3$.

$S \in N$ (Start Symbol).

Suppose $\alpha, \beta \in (N \cup \Sigma)^*$. β is derivable from α in one step $\alpha \xrightarrow{1} \beta$ if β can be obtained from α by replacing some occurence of a non-terminal A in α with γ where $A \to \gamma \in P$.

if $\exists\ \alpha_1, \alpha_2 \in (N \cup \Sigma)^*$, $\exists\ A \to \gamma \in P$ s.t

$$\alpha = \underline{\alpha_1 A \alpha_2} \quad \text{and} \quad \beta = \alpha_1 \gamma \alpha_2$$

$$\downarrow$$

$$\alpha_1 \gamma \alpha_2 = \beta$$

$\alpha \xrightarrow[G]{1} \beta$ - one step derivation

$\xrightarrow[G]{*}$: reflexive transitive closure of the relation $\xrightarrow[G]{1}$.

$\alpha \xrightarrow[G]{0} \alpha$ for all α.

$\alpha \xrightarrow[G]{n+1} \beta$ if $\exists \gamma$ s.t $\alpha \xrightarrow[G]{n} \gamma$ and $\gamma \xrightarrow[G]{1} \beta$.

$\alpha \xrightarrow[G]{*} \beta$ if $\alpha \xrightarrow[G]{n} \beta$ for some $n \geq 0$.

Language generated by G.

$$L(G) = \{ x \in \Sigma^* \mid S \xrightarrow[G]{*} x \}$$

$B \subseteq \Sigma^*$ is a Context free language (CFL) if

$B = L(G)$ for some CFG G.

$Z = \{a^n b^n \mid n \geq 0\}$ - is a CFL.

CFG. $G = (N, \Sigma, P, S)$ $N = \{S\}$, $\Sigma = \{a, b\}$.

$P = \{S \to aSb, S \to \epsilon\}$.

$L(G) = Z$.

$a^3 b^3$: $S \xrightarrow[G]{1} aSb \xrightarrow[G]{1} aaSbb$

$\xrightarrow[G]{1} aaaSbbb \xrightarrow[G]{1} aaabbb$.

By induction on n: show that $S \xrightarrow[G]{n+1} a^n b^n$

$\Rightarrow$ all strings of the form $a^n b^n \in L(G)$.

Conversly, the only strings in $L(G)$ are of the form $a^n b^n$ - induction on the length of the derivation