Computing concepts for Compiler Design

Design of Lexical Analyzer

Concepts and Notations

- **Set**: An unordered collection of unique elements

  S_1 = { a, b, c } S_2 = { 0, 1, …, 19 } empty set: $\varnothing$

  membership: $x \in S$ union: $S_1 \cup S_2$ = { a, b, c, 0, 1, …, 19 }

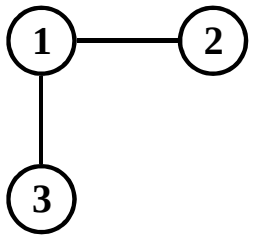
  universe of discourse: U subset: $S_1 \subset U$

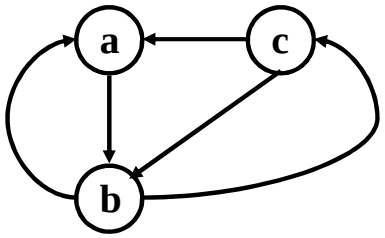
  complement: if U = { a, b, …, z }, then S_1' = { d, e, …, z } = $U - S_1$

- **Alphabet**: A finite set of symbols
  - Examples:
    - Character sets: ASCII, ISO-8859-1, Unicode
    - S_1 = { a, b } S_2 = { Spring, Summer, Autumn, Winter }
- **String**: A sequence of zero or more symbols from an alphabet
  - The empty string: ε

Concepts and Notations

- Language: A set of strings over an alphabet
  - Also known as a formal language; may not bear any resemblance to a natural language, but could model a subset of one.
  - The language comprising **all** strings over an alphabet $\sum$ is written as: $\sum^*$
- Graph: A set of nodes (or vertices), some or all of which may be connected by edges.
  - An example:
  - A directed graph example:





Regular Expression

- It is a tool to express language in the form of expression.
- RE uses primitive operators for expressing language.
- The three operators used are: Union, Concatenation, Kleene Star.
- Rules for Regular Expression:
  - Every letter of alphabet set represents regular expression.
  - ϕ is regular expression.
  - ε is regular expression.
  - If r1 and r2 are regular expressions then (r1), (r2), (r1 . r2) (r1+r2) ($r1^*$) ($r2^*$) will be denoted as regular expression.
  - Nothing else is regular expression.

Examples on regular expressions

Language	Regular Expression	Remarks
L = {a^n bm | n+m is even}	(aa)*(bb)* + (aa)*a(bb)*b	Both n and m are even Both n and m are odd
L = {All set of strings of (0,1) with even 0s followed by odd 1s} L = {All set of strings of (0,1) with even 0s or odd 1s}	(00)* (11)* 1 (00)* + (11)* 1	
L = {a,b} No two consecutive letters are same	Two possibilities: b (ab)* a a (ba)* b	
L = {W ε (a,b) with length(w)=even}	((aa) (ab) (ba) (bb))*	
L = {W ε (a,b)*, such that W contains even number of a's and odd number of b's and	L = L1 UNION L2 AND if L1 and L2 are regular, then L is regular.	

Examples: Regular Expressions

1. 0^*10^*, $L(0^*10^*) = \{w \mid w$ contains exactly a single 1$\}$
2. $\Sigma^*1\Sigma^*$, $L(\Sigma^*1\Sigma^*) = \{w \mid w$ contains at least one 1$\}$
3. $\Sigma^*001\Sigma^*$, $L(\Sigma^*001\Sigma^*) = \{w \mid w$ contains 001 as a substring$\}$
4. $(\Sigma\Sigma)^*$, $L(\Sigma\Sigma)^* = \{w \mid w$ is a string of even length$\}$
5. $(\Sigma\Sigma\Sigma)^*$, $L(\Sigma\Sigma\Sigma)^* = \{w \mid$ the length of w is a multiple of three$\}$
6. $01 \cup 01$, $L(01 \cup 01) = \{01, 01\}$
7. $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$, $L(0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1) = \{w \mid w$ starts and ends with the same symbol$\}$
8. $(0 \cup \epsilon)1^*$, $L((0 \cup \epsilon)1^*) = \{01^* \cup 1^*\}$
9. $(0 \cup \epsilon)(1 \cup \epsilon)$, $L((0 \cup \epsilon)(1 \cup \epsilon)) = \{\epsilon, 0, 1, 01\}$

Examples:

Examples of Regular Languages

- $L = \{x \in \{0,1\}^* \mid |x| \text{ is even}\}$
  - Any string of even length can be obtained by concatenating strings length 2.
  - Any concatenation of strings of length 2 will be even
  - $L = \{00, 01, 10, 11\}^*$
  - Regular expressions describing L:
    - $(00 + 01 + 10 + 11)^*$
    - $((0 + 1)(0 + 1))^*$

Examples of Regular Languages

- $L = \{x \in \{0,1\}^* \mid x \text{ does not end in } 01\}$
  - If x does not end in 01, then either
    - $|x| < 2$ or
    - x ends in 00, 10, or 11
  - A regular expression that describes L is:
    - $\Lambda + 0 + 1 + (0 + 1)^*(00 + 10 + 11)$

Examples of Regular Languages

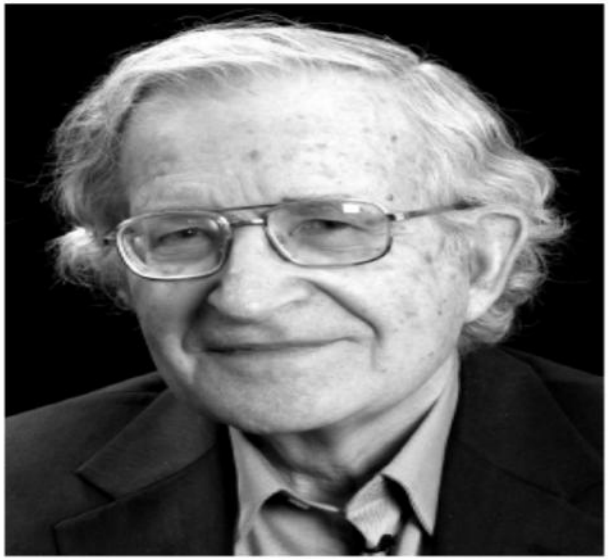
- $L = \{x \in \{0,1\}^* \mid x \text{ contains an odd number of 0s}\}$
  - Express x = yz
  - y is a string of the form $y=1^i01^j$
  - In z, there must be an even number of additional 0s or $z = (01^k01^m)^*$
  - x can be described by $(1^*01^*)(01^*01^*)^*$

Examples

- Let $\Sigma = \{a, b\}$
  - $a|b$ denotes $\{a, b\}$
  - $(a|b)(a|b)$ denotes $\{aa, ab, ba, bb\}$
    i.e., $(a|b)(a|b) = aa|ab|ba|bb$
  - a^* denotes $\{\epsilon, a, aa, aaa, ...\}$
  - $(a|b)^*$ denotes the set of all strings of a's and b's (including ϵ)
    i.e., $(a|b)^* = (a^*b^*)^*$
  - $a|a^*b$ denotes $\{a, b, ab, aab, aaab, aaaab, ...\}$

Context Free Grammars

Context-Free Grammars



Noam Chomsky
(linguist, philosopher, logician, and activist)

" A grammar can be regarded as a device that enumerates the sentences of a language. We study a sequence of restrictions that limit grammars first to Turing machines, then to two types of systems from which a phrase structure description of a generated language can be drawn, and finally to finite state Markov sources (finite automata). "

Chomsky Hierarchy of Grammar

Description	Type	Terminology
Un-restricted Grammar	Type – 0	No restriction on writing production rules.
Context Sensitive Grammar	Type – 1 aXb → aYb If X = surrounded by “a” and ”b” replace X with Y	Depending on context, non-terminals are expanded. Tag rules of HTML language
Context Free Grammar	Type -2	The production rule will only have one non—terminal on LHS Permitted rules A → a A → B A → any combination of terminal and non-terminals A → ε
Regular Grammar Restricted grammar	Type - 3	Permitted rules: A → aB or A → a or A → ε

Formal Definition of Context Free Grammar

- CFG is defined as: G = (V, T, P, S)
  - V = set of non-terminals
  - T = set of terminals
  - P = Production rules: of form A → γ where y = (VUT)*
  - γ can be ε or single terminal or single non-terminal symbol.
- γ can be any combination of terminal and non-terminals
- Why CFG:
- Since there is no restriction on right side of production rule, it is denoted as context free, i.e., ir-respective of surrounding, non-terminal symbol can be replaced by RHS if there exist production rule.

Design of CFG

- Example: 1: Balanced parenthesis grammar
  - S → (S)
  - S → SS
  - S → ε
- Example 2: Grammar for L = {$a^n b^n$ |n>=0}
  - S → aSb
  - S → ε

Design of CFG

- Example: 3: Even length palindrome
  - S → aSa
  - S → bSb
  - S → ε
- Example 4: Grammar for L = Equal number of a's and b'
  - S → aSb
  - S → bSa
  - S → SS
  - S → ε

Design of CFG

- Example: 4: L = {W ε (0,1)* | W contains three 1's}
  - S → X 1 X 1 X 1
  - X → 0X | 1X | ε
- Example 5: L = {W ε (0,1)* | W length is odd and middle symbol is 0}
  - S → 0S0 | 1S1 | 0S1 | 1S0
- Example 6: L = {$a^i b^j c^k$ | i+j = k}
  - S → aSc | X
  - X → bXc | ε

Write CFGs for the following languages:

1. Strings ending with a 0
2. Strings containing even number of 1's
3. palindromes over $\{0,1\}$
4. $L = \{a^i b^j : i \leq 2j\}$ or $L = \{a^i b^j : i < 2j\}$ or $L = \{a^i b^j : i \neq 2j\}$
5. $L = \{a^i b^j c^k : i = k\}$
6. $L = \{a^i b^j c^k : i = j\}$
7. $L = \{a^i b^j c^k : i = j + k\}$.
8. $L = \{w \in \{0,1\}^* : |w|_a = |w|_b\}$.

Reading assignment

- Read:
  - Cross Compiler
  - Example of cross compiler
  - Bootstrapping
  - Example of bootstrapping