**CS105:Quick Reference to Chapter 2** **Date:** 18 July, 2004
**Author:** Ragesh Jaiswal

## 1.1 Context Free Grammars

1. Context Free Grammar(CFG): More powerful method of describing languages. Describe certain features that have recursive structure.
2. Context Free Languages(CFL): The collection of language associated with context free grammars.
3. Pushdown Automata(PDA): A class of machines recognizing the context free languages.
4. Terminology for CFG: Substitution rules, productions, variable, terminals, start variable.
5. Derivation: A sequence of substitutions to obtain a string.
6. Parse Tree: Pictorial representation of the derivation.
7. Language of the Grammar: All strings generated from such derivation.
8. Formal definition of CFG: A context free grammar is a 4 tuple $(V, \Sigma, R, S)$, where
   (a) $V$ is a finite set called the variables
   (b) $\Sigma$ is a finite set, disjoint from $V$ called the terminals
   (c) $R$ is a finite set of rules, with each rule being a variable and a string of variables and terminals.
   (d) $S \in V$ is the start variable.
9. If the grammar generates the same string in several different ways, we say that the string is derived ambiguously in that grammar. If a grammar generates some string ambiguously we say that the grammar is ambiguous.
10. A string $\omega$ is derived ambiguously in context free grammar $G$ if it has two or more different leftmost derivations. Grammar $G$ is ambiguous if it generates some string ambiguously.
11. Push Down Automata: New type of computational model.
12. Non-determinism adds power to the capability of pushdown automata.
13. Formal definition of a PDA: A pushdown automaton is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where $Q, \Sigma, \Gamma$, and $F$ are all finite sets, and
   (a) $Q$ is a set of states
   (b) $\Sigma$ is the input alphabet

(c) $\Gamma$ is the stack alphabet

(d) $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to \mathcal{P}(Q \times \Gamma_\epsilon)$ is the transition function

(e) $q_0 \in Q$ is the start state, and

(f) $F \subseteq Q$ is the set of accept states.

14. A pushdown automata computes as follows. It accepts input string $\omega$ if $\omega$ can be written as $\omega_1, \omega_2, ..., \omega_m$, where each $\omega_i \in \Sigma_\epsilon$ and sequences of states $r_0, r_1, ..., r_m \in Q$ and strings $s_0, s_1, ..., s_m \in \Gamma^*$ exist that satisfy the next 3 conditions. The string $s_i$ represent the sequence of stack contents that $M$ has on the accepting branch of computation.

    (a) $r_0 = q_0$ and $s_0 = \epsilon$, This condition signifies that $M$ starts out properly, in the start state and with empty stack.

    (b) For $i = 0, ..., m-1$, we have $(r_{i+1}, b) \in \delta(r_i, \omega_{i+1}, a)$, where $s_i = at$ and $s_{i+1} = bt$ for some $a, b \in \Gamma_\epsilon$ and $t \in \Gamma^*$. This condition states that $M$ moves properly according to the state, stack and next input symbol.

    (c) $r_m \in F$. This states that an accept state occurs at the end of the input.

15. A language is context free if and only if some push down automaton recognizes it.

16. Pumping lemma for CFGs: If $A$ is a context-free languages, then there is a number $p$ (pumping length) where, if $s$ is any string in $A$ of length at least $p$, then $s$ may be divided into 5 pieces $s = uvxyz$ satisfying the conditions:

    (a) For each $i \geq 0$, $uv^ixy^iz \in A$

    (b) $|vy| > 0$, and

    (c) $|vxy| \leq p$.