Chapter 2

Context Free Languages and Push Down Automata

LEARNING OBJECTIVES

- *Context free grammar*
- *Context free language*
- *Ambiguity in context free grammars*
- *Removing ⍰-productions*
- *Removing unit productions*
- *Normal forms*
- *Chomsky's normal form*
- *Greiback normal form*
- *Closure properties of CFL's*
- *Push down automata*
- *PDAs accepting by final state and empty stack are equivalent*
- *Converting CFG to PDA*
- *Deterministic PDA*

Context Free Grammar

- A context free grammar (CFG) is a finite set of variables (non-terminals) each of which represents a language. The language represented by variables is described recursively in terms of each other. The primitive symbols are called terminals.
- The rules relating variables are called productions. A typical production states that the language associated with a given variable contains strings that are formed by concatenating strings from languages of certain other variables.
- CFG is a collection of three things;
  An alphabet Z of letters called terminals.
  A set of symbols called non-terminals, one of which is a start symbol, S.
  A finite set of productions of the form:
  One terminal $\rightarrow$ finite set of terminals and/or non-terminals.
- A CFG is defined as: $G = (V, T, P, S)$

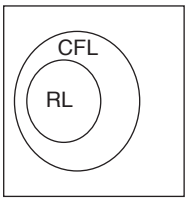
  Where

  $V \rightarrow$ Finite set of variables (non-terminals)
  $T \rightarrow$ Finite set of terminals (symbols)
  $P \rightarrow$ Finite set of productions, each, production is of the form, $A \rightarrow \alpha, A \in V, \alpha \in (V \cup T)^*$
  $S \rightarrow$ Start symbol

Context Free Language (CFL)

The language generated by CFG is a set of all strings of terminals that can be produced from start symbols, using the productions as substitutions. A language generated by a CFG is called context free language (CFL).

Note: Every regular grammar is context free, so a regular language (RL) is also context free.
Family of RL's is proper subset of CFL's.
i.e., RL $\subset$ CFL

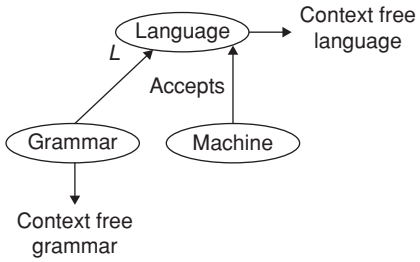


Solved Examples

Example 1: What is the language that is generated by CFG, $G = S \rightarrow AB|A \rightarrow +/-|B \rightarrow CB/C|C \rightarrow 0/1/2/ \ldots 9$.
(A) Set of all rational numbers
(B) Set of all integers
(C) Set of all natural numbers
(D) Set of all complex numbers

Solution: (B)
$S \rightarrow AB|A \rightarrow +/-|B \rightarrow CB/C|C \rightarrow 0/1/2/ \ldots 9$
Consider-18 (integer)

$S \rightarrow AB$
$\rightarrow -B$
$\rightarrow -CB$
$\rightarrow -1B$
$\rightarrow -18$



Ambiguity in Context Free Grammars

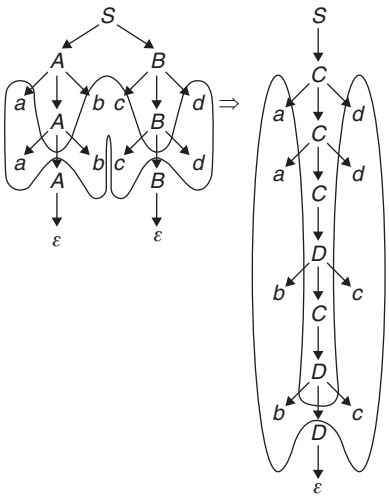
A CFG, G is called ambiguous if there is $w \in L(G)$ such that w has (at least) two different parse trees with respect to G.

Example 2: The language, $L = \{a^n\ b^n\ c^m\ d^m / n \geq 0,\ m \geq 0\} \cup \{a^n b^m c^m d^n / n \geq 0,\ m \geq 0\}$ is designed in CFG, G. The Grammar is
(A) Ambiguous
(B) Unambiguous
(C) Cannot be determined
(D) None of above

Solution: (A)
CFG G for given language L is:

$S \rightarrow AB|C$
$A \rightarrow aAb|\varepsilon$
$B \rightarrow cBd|\varepsilon$
$C \rightarrow aCd|$D
$D \rightarrow bDc|\varepsilon$

It's an inherently ambiguous grammar.
Consider string, aabbccdd



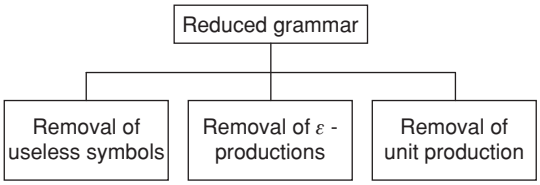
Note:
- A context free language with property that all grammars that generate it are ambiguous is inherently ambiguous.
- Inherently ambiguous grammars cannot convert to unambiguous grammars.

Minimization of Context Free Grammar

- Grammar may consist of some extra symbols (non-terminals). Having extra symbols unnecessarily increases the length of grammar.
- Simplification of grammar means reduction of grammar.

The properties of reduced grammar are:

1. Each variable (non-terminal) and each terminal of G appears in the derivation of some word in L.
2. There should not be any production as $X \rightarrow Y$ where X and Y are non-terminals.
3. If ε is not in language L, then there need not be production $X \rightarrow \varepsilon$.



Removal of Useless Symbols

- Any symbol is useful when it appears on right hand side, in the production rule and generates some terminal string. If no such derivation exists, then it is supposed to be a useless symbol.
- A symbol P is useful, if there exists some derivation

$$S^* \Rightarrow \alpha PB \text{ and } \alpha PB \overset{*}{\Rightarrow} W$$

Then P is said to be useful symbol.

Example 3: A grammar G', is generated by removing useless symbols from G defined below. The obtained G' contains productions:

$S \rightarrow aA|bB$
$A \rightarrow aA|a$
$B \rightarrow bB$
$D \rightarrow ab|Ea$
$E \rightarrow aC|d$

(A) $S \rightarrow aA$
$A \rightarrow aA|a$

(B) $S \rightarrow aS|bA|C$
$A \rightarrow a$
$C \rightarrow aCd$

(C) $S \rightarrow aA|bB$
$A \rightarrow aA|a$
$B \rightarrow bB$

(D) Cannot remove useless symbols

Solution:
$S \to aA \to aaA \to aaaA \to aaaa$ ✓
$B \to bB \to bbB \to bbbB \to bbbbB$ ..... (string cannot be generated)
$\therefore$ B is useless
D and E cannot be generated from 'S'. So, eliminate. Hence G' contains
$\therefore$ $S \to aA$
$A \to aA|a$

Removing ε-Productions

A production of the form $A \to \varepsilon$ is called an ε-production. If A is a non-terminal and $A \to (^*)\ \varepsilon$, then A is called a 'nullable non-terminal'. So eliminate such productions without changing meaning of grammar.

Example 4: The grammar, G is given below. The CFG generated after eliminating ε-production is:

$S \to ABC$
$A \to BC|a$
$B \to bAC|\in$
$C \to cAB|\in$

(A) $S \to ABC|AB|BC|CA$
$A \to BC|B|C$
$B \to bAC|bA|bC$
$C \to cAB|cA|cB$

(B) $S \to ABc$
$A \to BC$
$B \to bAC$
$C \to cAB$

(C) $S \to ABC|BC|AC|AB|A|B|C$
$A \to BC|B|C|a$
$B \to bAC|bA|bC|b$
$C \to cAB|cA|cB|c$

(D) None of these

Solution (C)
$B \to \in, C \to \in$
$\Rightarrow$ $A \to \in$
$\therefore$ Remove ε-productions and obtained CFG is Choice (C).

Removing Unit Productions

- A production of form $A \to B$, where A and B are both non-terminals, is called a 'unit production'.
- Presence of unit production in a grammar increases the cost of derivation.

Example 5: The total number of productions obtained by removing unit production from the Grammar,

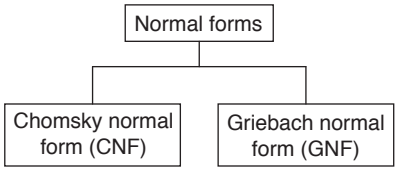
$A \to PQ$
$P \to 0$
$Q \to R|1$
$R \to S$
$S \to W|1R$
$W \to 2|P1$

(A) 9 (B) 2
(C) 3 (D) 5

Solution: (A)
$A \to PQ$
$Q \to R \to S \to W \to 2$
$\Rightarrow$ $Q, R, S \to$ Unit production
$A \to PQ$
$P \to 0$
$Q \to R|1$
$\downarrow$
$\Rightarrow$ $Q \to 2|P1|1R|1$ (substitute the production of R, S, W)
$\therefore$ $A \to PQ$
$P \to 0$
$Q \to 2|\ P1|\ 1R|\ 1$
$R \to 2|\ P1|\ 1R$
$\therefore$ 9 – Productions

Normal Forms

- It is necessary to have a grammar in some specific form so, grammar normalization is needed.



That is, There should be fixed number of terminals and non-terminals, in CFG.

Chomsky's Normal Form (CNF)

- A context free grammar (CFG), $G = (V, \Sigma, R, S)$ is said to be in CNF, if and only if every rule in R is of one of the following forms
  1. $A \to a$, for some $A \in V$ and some $a \in \Sigma$
  2. $A \to BC$, for some $A \in V$ and $B, C \in V \cup \{S\}$
  3. $S \to \varepsilon$
- Every rule either replaces a variable by a single character or by a pair of variables except the start symbol and the only rule that can have the empty word as it's right hand side must have start symbol as it's left hand side.

Note: Every parse tree for a grammar in CNF must be a binary tree and the parse tree for any non-empty word cannot have any leaves labeled with ε in it.

Transforming of a grammar to CNF

- In order to construct the grammar G in CNF that is equivalent to a given grammar G, first identify how exactly G can violate the rules for a CNF. Since CNF only restricts the rules in G, see only at R. The 'bad' cases of rules are:
- $A \to uSv$ where $A \in V$ and $u, v \in (V \cup \Sigma)*$. The start symbol must not appear on the right-hand side of any rule. This is called 'start symbol rule'.

- **To remove 'start symbol rule',** add a new symbol, so make it the start symbol in new grammar G_1, and add the single rule $S_0 \to S$ to R to get the rules for G_1. Since S_0 does not appear in any rules, the new grammar has no start symbol rules.
  - $A \to \varepsilon$ where $A \in V \cup \{S\}|$. The only symbol that can be replaced by the word is start symbol. This is called 'ε-rules'.
- **To remove 'ε-rules',** identify all variables that can yield the empty string, either directly or indirectly.

  These variables are 'nullable'. Remove all direct rules $A \to \varepsilon$ from the grammar and fix up the grammar by removing all occurrences of nullable variables from the right hand sides of all rules.

  $A \to B$ where $A, B \in V$. The only rules involving variables on the right-hand side must have exactly two of them. This is called 'unit rules'.
- **To remove 'Unit rules',** identify a set of unit pairs. These are pairs of symbols (A, B), where $A \overset{*}{\Rightarrow} B$. Then remove all unit rules by copying right-hand sides. If there is a rule $A \to B$, (A, B) is a unit pair. Then, if there is a rule $B \to W$, derive W from A by $A \to B$, $B \to W$. To remove the unit rule and still generate an equivalent grammar, add the right-hand side W to the rules for A directly, $A \to W$.

  $A \to W$ where $A \in V$, $W \in (V \cup \Sigma)^*$ and W contains at least one character and at least one variable. The only rules where character appear on right-hand side must have exactly one character as right-hand side. This is called 'mixed rules'.
- **To remove 'mixed rules',** Let $A \to W \in R_3$ is a mixed rules. Then write W as $W = V_0\, C_1 V_1 \ldots V_{n-1} C_n V_n$, where $C_i \in \Sigma$ are occurrences of characters, and the $V_i \in V^*$ are strings of only variables. Then add a new symbol, C_i to V_4 for every character C_i and add the rules $C_i \to c_i$ to R_4. Finally define $W^1 := V_0\, C_1\, V_1 \ldots V_{n-1}\, C_n\, V_n \in V^*$ and add rules $A \to W'$ to R_4. If the rule $A \to W$ is part of the derivation for some word, replace that single rule by applying rule $A \to W'$ first and then replacing all C_i by c_i using their respective rules.

  $\boxed{A \to w}$ Where $A \in V$ and $W \in (V \cup \Sigma)^*$ with $|w| > 2$. Rules must have one symbol (character) or two variables (two variables as right hand side). These are called long rules.
- **To remove 'long rules',** Let $A \to B_1 \ldots B_n$ be a long rule, i.e., $n > 2$. B_i is all variables. Break up every single long rule, into several 'short' rules, by introducing new 'helper variables' and splitting right hand side from left to right: add new symbols $A_1, \ldots A_{n-2}$ to set of variables and add following rules to R_5: $A \to B_1 A_1, A_1 \to B_2 A_2, \ldots A_{n-2} \to B_{n-1} B_n$.

Example 6: Consider grammar, $G = S \to ASB$, $A \to aAS|a|\varepsilon$, $B \to SbS|A|bb$. The CNF generated contains ____ non-terminals.

(A) 5 (B) 6
(C) 9 (D) 11

Solution: (C)

Add new start state:

$S_0 \to S$
$S \to ASB$
$A \to aAS|a|\varepsilon$
$B \to SbS|A|bb$

Eliminate ε-rules

$A \to \varepsilon$:
$S_0 \to S$
$S \to ASB|SB$
$A \to aAS|a|aS$
$B \to SbS|A|bb|\varepsilon$

Eliminate $B \to \varepsilon$:

$S_0 \to S$
$S \to ASB|SB|S|AS$
$A \to aAS|a|aS$
$B \to SbS|A|bb$

Remove Unit rules:

$B \to A$:
$S_0 \to S$
$S \to ASB|SB|S|AS$
$A \to aAS|a|aS$
$B \to SbS|bb|aAS|a|aS$

$S \to S$:
$S_0 \to S$
$S_0 \to ASB|SB|AS$
$A \to aAS|a|aS$
$B \to SbS|bb|aAS|a|aS$

$S_0 \to S$:
$S_0 \to ASB|SB|AS$
$S \to ASB|SB|AS$
$A \to aAS|a|aS$
$B \to SbS|bb|aAS|a|aS$

Replace rules which have more than two symbols:

$S_0 \to ASB$: $S_0 \to AU_1$ and $U_1 \to SB$
$\therefore S_0 \to AU_1|SB|AS$
$S \to AU_1|SB|AS$
$A \to aAS|a|aS$
$B \to SbS|bb|aAS|a|aS$
$U_1 \to SB$
$A \to aAS \Rightarrow A \to aU_2$ and $U_2 \to AS$ and $B \to SbS$
$\Rightarrow B \to SU_3$ and $U_3 \to bS$
$\therefore S_0 \to AU_1|SB|AS$
$S \to AU_1|SB|AS$
$A \to aU_2|a|aS$
$B \to SU_3|bb|aU_2|a|aS$
$U_1 \to SB$
$U_2 \to AS$
$U_3 \to bS$

Eliminate rules which have terminals and variables or two terminals.

Let $V_1 \to a$, $V_2 \to b$
$\therefore S_0 \to AU_1|AS|SB$
$S \to AU_1|SB|AS$
$A \to V_1U_2|a|V_1S$

$B \rightarrow SU_3|V_2V_2|V_1U_2|a|V_1S$
$U_1 \rightarrow SB$
$U_2 \rightarrow AS$
$U_3 \rightarrow V_2S$
$V_1 \rightarrow a$
$V_2 \rightarrow b$
$\therefore$ Nine non-terminals.

Greiback Normal Form (GNF)

- A CFG, $G = (V, T, R, S)$ is said to be in GNF, if every production is of form $A \rightarrow a\alpha$ where $a \in T$, $\alpha \in V^*$, i.e., α is a string of zero or more variables.
- Left recursion in R can be eliminated by following schema: If $A \rightarrow A\alpha_1 |A\alpha_2| \ldots |A\alpha_r|\beta_1|\beta_2| \ldots |\beta_s$, then replace the above rules by
  (i) $A \rightarrow \beta_i|\beta_iZ, 1 \le i \le s$
  (ii) $Z \rightarrow \alpha_i| \alpha_iZ, 1 \le i \le r$
- If $G = (V, T, R, S)$ is a CFG, then another CFG, $G_1 = (V_1, T, R_1, S)$ can be constructed in GNF $\exists$ $L(G_1) = L(G) - \{\varepsilon\}$.

The step wise algorithm is as follows:

1. Eliminate null production, unit productions and useless symbols from the grammar G and then construct a $G^1 = (V^1, T, R^1, S)$ in CNF generating the language $L(G^1) = L(G) - \{\varepsilon\}$.
2. Rename the variables like $A_1, A_2, \ldots A_n$ starting with $S = A_1$.
3. Modify the rules in R^1, so that if $A_i \rightarrow A_j\gamma \in R^1$ then $j > i$.
4. Starting with A_1 and proceeding to A_n, can be obtained as:
   (a) Assume that productions have been modified so that for $1 \le i \le k$, $A_i \rightarrow A_j\ \gamma \in R^1$ only if $j > i$
   (b) If $A_k \rightarrow A_j\gamma$ is a production with $j < k$, generate a new set of productions substituting for A_j, the body of each A_j production.
   (c) Repeating (b) atmost $k - 1$ times, obtains rules of the form $A_k \rightarrow A_p\gamma, p \ge k$.
   (d) Replace rules $A_k \rightarrow A_k\gamma$ by removing left-recursion.
5. Modify the $A_i \rightarrow A_j\gamma$ for $i = n - 1, n - 2, \ldots 1$ in desired form at same time change z production rules.

Example 7: A grammar G is defined with rules $S \rightarrow XA|BB$, $B \rightarrow b|SB$, $X \rightarrow b$, $A \rightarrow a$. The normalized GNF of G contains ____ productions.

(A) 17 (B) 19
(C) 5 (D) 16

Solution: (B)

1. The Grammar, G is already in CNF.
2. Re-label with variables
   S with A_1
   X with A_2
   A with A_3
   B with A_4
   Grammar, G now is:
   $A_1 \rightarrow A_2A_3|A_4A_4$
   $A_4 \rightarrow b|A_1A_4$
   $A_2 \rightarrow b$
   $A_3 \rightarrow a$
3. Identify all productions which do not conform to any of the types listed below:
   $A_i \rightarrow A_j\, x_k \exists\, j > i$
   $Z_i \rightarrow A_j\, x_k \exists\, j \le n$
   $A_i \rightarrow a\, x_k \exists\, x_k \in V^*$ and $a \in T$
4. $A_4 \rightarrow A_1\, A_4 \ldots$ identified
5. $A_4 \rightarrow A_1\, A_4|b$
   To eliminate A_1, use substitution rule, $A_1 \rightarrow A_2\, A_3|A_4A_4$
   $\therefore \quad A_4 \rightarrow A_2\, A_3A_4|A_4A_4A_4|b$
   Substitute $A_2 \rightarrow b$
   $\therefore \quad A_4 \rightarrow b\, A_3A_4|A_4A_4A_4|b$
   $A_4 \rightarrow A_4A_4A_4$ is left recursive. So, remove left recursion i.e., $A_4 \rightarrow b\, A_3A_4|b|bA_3A_4Z|bZ$
   $Z \rightarrow A_4A_4|A_4\, A_4Z$
6. Now, $G = A_1 \rightarrow A_2\, A_3|A_4A_4$
   $A_4 \rightarrow b\, A_3A_4|b|b\, A_3A_4\, Z|b\, Z$
   $Z \rightarrow A_4\, A_4|A_4A_4Z$
   $A_2 \rightarrow b$
   $A_3 \rightarrow a$
7. A_1, Z are not in GNF. So,
   For $A_1 \rightarrow A_2A_3|A_4A_4$:
   Substitute for A_2 and A_4 to convert it to GNF
   $A_1 \rightarrow b\, A_3| b\, A_3\, A_4A_4|b\, A_4|b\, A_3\, A_4Z\, A_4|b\, Z\, A_4$
   For $Z \rightarrow A_4\, A_4|A_4\, A_4Z$
   substitute for A_4 to convert it to GNF
   $Z \rightarrow b\, A_3\, A_4A_4|b\, A_4|b\, A_3\, A_4Z\, A_4|b\, Z\, A_4|b\, A_3\, A_4A_4$
   $Z|b\, A_4Z|b\, A_3A_4\, Z\, A_4Z|b\, Z\, A_4\, Z$
   $\therefore$ Final GNF is:
   $A_1 \rightarrow b\, A_3|b\, A_3A_4A_4|b\, A_4|\ b\, A_3A_4Z\, A_4|b\, Z\, A_4$
   $A_4 \rightarrow b\, A_3\, A_4|b|b\, A_3\, A_4Z|b\, Z$
   $A_2 \rightarrow b$
   $A_3 \rightarrow a$
   $Z \rightarrow b\, A_3\, A_4A_4|b\, A_4|b\, A_3A_4\, Z\, A_4|b\, Z\, A_4|b\, A_3\, A_4\, A_4$
   $Z|\ b\, A_4\, Z|b\, A_3\, A_4\, Z\, A_4Z|b\, Z\, A_4\, Z$
   $\therefore$ 19 productions.

Pumping Lemma for Context Free Languages

Let 'L' be context free language. There exists some integer, m $\exists$ $\forall$w in L, with $|w| \ge m$, $w = uvxyz$ with $|vxy| \le m$ and $|vy| \ge 1$ $\exists$ $u\, v^i\, x\, y^i\, z \in L$ $\forall$ $i = 0, 1, 2, 3, \ldots$

Note: Pumping lemma is used to show that a language is Not context free.

Example 8: The language $\{a^n\, b^m\, c^n\, d^{(n+m)}: m, n \ge 0\}$ is

(A) Regular
(B) Context free but not regular
(C) Neither context free nor regular
(D) Cannot be determined

Solution: (C)
$L = \{a^n b^m c^n d^{(n+m)}: m, n \geq 0\}$
Clearly, L is not regular because, number of a's and number of b's must be known to compute number of d's.
'L' is not context free because, Let $w = a^M b^M c^M d^{2M}$. Clearly neither v nor y can cross regions and include more than one letter, since if that happened; letters obtained will be out of order when pumped.

So, consider cases, where v and y fall within a single region. Consider 4-regions corresponding to a, b, c and d.
(1, 1) → change number of a's and they won't match c's any more.
(1, 2) → If v is not empty, change a's and they won't match with c's. If y is non-empty, number of b's changed won't have right number of d's.
(1, 3), (1, 4) → ruled out. ∵ $|v\,x\,y| \leq M$
(2, 2) → Change number of b's and they won't match right number of d's.
(2, 3) → If v is non-empty, change number of b's without changing number of d's. If y is not empty, change c's and they'll no longer match a's.
(2, 4) → ruled out ∵ $|v\,x\,y| \leq M$
(3, 3) → Change number of c's and they won't match a's.
(3, 4) → If v is not empty change c's and they won't match a's. If y is not empty, change d's without changing b's.
(4, 4) → change d's without changing a's or b's.
∴ L is not context free.

Closure Properties of CFL's

1. **CFL's are closed under union:** For CFL's L_1, L_2 with CFG's G_1, G_2 and start variables S_1, S_2. The grammar of Union $L_1 \cup L_2$ has new start symbol S and additional production $S \to S_1 | S_2$
2. **CFL's are closed under concatenation:** For CFL's L_1, L_2 with CFG's G_1, G_2 and start variables S_1, S_2. The grammar of concatenation L_1L_2 has new start variables S and additional production: $S \to S_1 S_2$
3. **CFL's are closed under star operation:** For CFL L, with CFG G and start variable S. The grammar of the start operation L^* has new start variable S_1 and additional production:
$$S_1 \to S\,S_1 | \varepsilon$$
4. **CFL's are not closed under intersection:** If L_1, L_2 are two context free languages, $L_1 \cap L_2$ not necessarily be context free.
5. **CFL's are not closed under complement:** If L is context free language, $\overline{L}$ not necessarily be context free.
6. **Intersection of CFL's and regular language: (regular closure):** If L_1 is a CFL and R_2 is a regular language then $L_1 \cap L_2$ is a CFL.

Example 9: The language, $L_1 = \{a^n b^n : n \geq 0\}$ and $L_2 = \{a^{100} b^{100}\}$. The relation $L_1 \cap \overline{L_2}$ is _____
(A) Regular
(B) Context free
(C) Regular but not context fee
(D) Cannot be determined

Solution: (B)
$L_1 = \{a^n b^n : n > 0\}$ is context free
$L_2 = \{a^{100} b^{100}\}$ is regular
$\overline{L_2} = \{(a+b)^*\} - \{a^{100}b^{100}\}$ is regular
$\{a^n b^n\}$ context free
$\overline{L_2} = \{(a+b)^*\} - \{a^{100}b^{100}\}$ is regular
$\{a^n b^n\} \cap \overline{L_2}$ → context free
$\{a^n b^n\} \cap \overline{L_2} = \{a^n b^n : n \neq 100,\ n \geq 0\} = L$ is context free:

Table 1 *Comparing Regular and Context free Languages:*

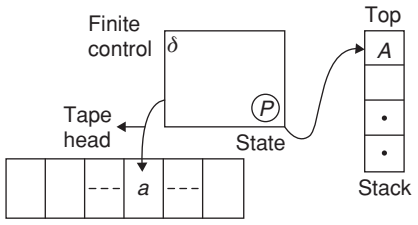
Regular Language	CFL
Regular expression or regular grammar	Context free grammar
Recognize the language	Parses the language
These are DFSA's	These are NDPDA's
Minimize FSA's	Find deterministic grammar.
Closed under: Concatenation Union Kleen star Complement Intersection	Closed under: Concatenation Union Kleen star

Push Down Automata (PDA)

A push down automata is merely a finite automata with a stack added to it.

PDA is used to generate context free language.

The stack allows for unbounded memorization.



Input tape: The tape is divided into finitely many cells. Each cell contains a symbol in an alphabet, Σ.
Stack: The stack head always scans the top symbol of the stack. It performs two basic operations.

- Push: Add a new symbol at the top
- Pop: Read and remove the top symbol

Tape head: The head scans at a cell on the tape and can read a symbol on the cell. In each move, the head can move to the right cell.

Finite control: The finite control has finitely many states which form a set Q. For each move, the state is changed according to the evaluation of transition function.

A PDA is defined as: $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$
Where Q: set of States
Σ: Input alphabet
Γ: Stack symbol
δ: Transition function
q_0: Start state
z_0: Initial stack top symbol
F: Final/accepting states
Transition functions δ: $Q \times \Gamma \times \Sigma \Rightarrow Q \times \Gamma$
Q: Old state
Γ: Stack top
Σ: Input symbol
Q: New state, Γ: New stack top

PDA's instantaneous description (IDs): A PDA has a configuration at any given instance: (q, w, y)

$q \rightarrow$ current state
$w \rightarrow$ remainder of input (i.e., unconsumed part)
$y \rightarrow$ current stack contents as a string from top to bottom of the stack.

If $\delta(q, a, x) = \{P, A\}$ is a transition, then following are also true:

- $(q, a, x) \vdash (P, \varepsilon, A)$
- $(q, aw, xB) \vdash (p, w, AB)$

Note: 1. $\rightarrow$: Turnstile notation and represents one move.
2. $\vdash^*$: represents sequence of moves.

Principles about IDs:

1. If for a PDA, $(q, x, A) \vdash^* (p, y, B)$, then for any string $w \in \Sigma^*$ and $\gamma \in \Gamma^*$, it is also true that: $(q, xw, A\gamma) \vdash^* (p, yw, B\gamma)$
2. If for a PDA, $(q, xw, A) \vdash^* (p, yw, B)$, then it is also true that: $(q, x, A) \vdash^* (p, y, B)$

Acceptance by PDA: There are two types of PDAs that one can design:

- Those that accept by final state or
- Those that accept by empty stack

PDAs that accept by final state: For a PDA, P, the language accepted by P, denoted by $L(P)$ by final state, is:

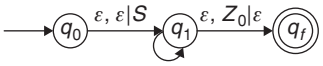
$$\{w | (q_0, w, z_0) \vdash^* (q, \in, A)\} \exists\, q \in F$$

PDAs that accept by empty stack: For a PDA P, the language accepted by P, denoted by $N(P)$ by empty stack, is:

$$\{w | (q_0, w, z_0) \vdash^* (q, \varepsilon, \varepsilon)\}, \text{ for any } q \in Q.$$

Example 10: Consider the grammar $S \rightarrow aTb | b$, $T \rightarrow Ta | \varepsilon$. The PDA constructed contains ____ states.
(A) 4 (B) 3 (C) 5 (D) 2

Solution: (B)
$S \rightarrow aTb | b$
$T \rightarrow Ta | \varepsilon$



$\varepsilon, S | aTb$
$\varepsilon, T | Ta$
$\varepsilon, S | b$
$\varepsilon, T | \varepsilon$
$a, a | \varepsilon$
$b, b | \varepsilon$

Let $S \rightarrow q_0$, $T \rightarrow q_1$
Consider string "aab" $S \rightarrow aTb \rightarrow aTab \rightarrow aab$
$\delta(q_0, aab, z_0) \vdash \delta(q_0, \varepsilon\, aab, z_0)$
$\vdash \delta(q_1, aab, q_0 z_0)$
$\vdash \delta(q_1, aab, aTb\, z_0)$
$\vdash \delta(q_1, ab, Tbz_0)$
$\vdash \delta(q_1, ab, aTbz_0)$
$\vdash \delta(q_1, b, Tbz_0)$
$\vdash \delta(q_1, b, \in bz_0)$
$\vdash \delta(q_1, b, bz_0)$
$\vdash \delta(q_1, \in, z_0)$
$\vdash \delta(q_f, \in) \rightarrow$ acceptance

PDAs accepting by final state and empty stack are equivalent:
$P_F \rightarrow$ PDA accepting by final state,
$P_F = (Q_F, \Sigma, \Gamma, \delta_F, q_0, z_0, F)$
$P_N \rightarrow$ PDA accepting by empty stack
$P_N = (Q_N, \Sigma, \Gamma, \delta_N, q_0, z_0)$

- For every P_N, $\exists\, P_F \exists\, L(P_F) = L(P_N)$
- For every P_F, $\exists\, P_N \exists\, L(P_N) = L(P_F)$

Converting CFG to PDA

The PDA simulates the left most derivation on a given w, and upon consuming it fully it either arrives at acceptance (by empty stack) or non-acceptance.

The steps to convert CFG to PDA are:

1. Push right hand side of the production on to stack, with left most symbol at the stack top.
2. If stack top is the left most variable, then replace it by all its productions (each possible substitution will represent a distinct path taken by non-deterministic PDA (NPDA).
3. If stack top has a terminal symbol and if it matches with the next symbol in the input string, then pop it. Follow from step-1 again to complete all productions.

Example 11: The CFG, G of a language L is $S \rightarrow AB$, $A \rightarrow aAb | \varepsilon$, $B \rightarrow cB / \varepsilon$. The PDA generated by G contains ____ states.
(A) 5 (B) 4 (C) 3 (D) 1

Solution: (C)

$S \to AB$
$A \to aAb|\varepsilon$
$B \to cB|\varepsilon$
$\Rightarrow \delta(q_0, w, S) = (q_1, AB)$
$\delta(q_1, w, A) = (q_1, aAb)$
$\delta(q_1, \varepsilon, A) = \delta(q_1, \varepsilon)$
$\delta(q_1, w, B) = (q_1, cB)$
$\delta(q_1, \varepsilon, B) = \delta(q_2, \varepsilon) \to$ accept
$\therefore \{q_0, q_1, q_2\}$ 3-states.

Converting a PDA into a CFG

Given: $G = (V, T, P, S)$ Initial stack symbol (S) same as start variable in grammar
Output: $P_N = (\{q\}, T, V \cup T, \delta, q, S)$, where δ is

- If q_0 is start state in PDA and q_n is final state of PDA then $[q_0, z, q_n]$ becomes a start state of CFG. Here z represents stack symbol.
- The production rule for the ID of the form $\delta(q_i, a, z_0) = (q_{i+1}, z_1, z_2)$ can be obtained as:

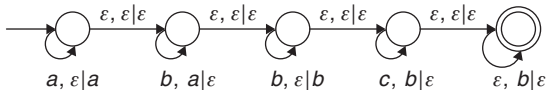
$$\delta(q_i, z_0, q_{i+k}) \to a(q_{i+1}, z_1, q_m)(q_m, z_2, q_{i+k})$$

Where q_{i+k}, q_m represents the intermediate staes, z_0, z_1, z_2 are stack symbols and a is input symbol.

- The production rule for the ID of the form $\delta(q_i, a, z_0) = (q_{i+1}, \varepsilon)$ can be converted as

$$(q_i, z_0, q_{i+1}) \to a$$

Example 12: The PDA, P for language L is generated as:



The CFG for P is:

(A) $S \to S_1S_2$
$S_1 \to aS_2b$
$S_2 \to c|\varepsilon$

(B) $S \to S_1bc$
$S_1 \to a|\varepsilon$

(C) $S \to S_1S_2$
$S_1 \to aS_1b|\varepsilon$
$S_2 \to bS_2|bS_2c|\varepsilon$

(D) $S \to aS_1c$
$S_1 \to b|\varepsilon$

Solution: (C)
The language, L generated by given PDA is

$$L = \{a^n b^n b^m c^p : m \geq p \text{ and } n, p \geq 0\}$$

It can be generated by following rules:

$$S \to S_1S_2$$

$S_1 \to aS_1b|\varepsilon \to S_1$ generates $a^n b^n$
$S_2 \to bS_2|bS_2c|\varepsilon \to S_2$ generates $b^m c^p$

Deterministic PDA (Deterministic CFL)

$$\left\{\begin{array}{c}\text{Deterministic}\\ \text{context free}\\ \text{languages}\\ \text{(DPDA)}\end{array}\right\} \subseteq \left\{\begin{array}{c}\text{Context-free}\\ \text{Languages}\\ \text{PDAs}\end{array}\right\}$$

- Every DPDA is also a PDA.
- A context free language 'L' accepted by PDA may or may not be accepted by DPDA.

A PDA, $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ is deterministic if there is no configuration for which M has choice of more than one move. That is, it must satisfy the following conditions:

1. For any $q \in Q$, $a \in \Sigma\varepsilon$ and $s \in \Gamma\ \varepsilon$, the set $\delta(q, a, s)$ has almost one element. (Doesn't allow two or more transitions from same state).
2. For any $q \in Q$, and $s \in \Gamma\varepsilon$, if $\delta(q, \varepsilon, s) \neq \phi$, then $\delta(q, a, s) = \phi$ for every $a \in \Sigma$ and $\delta(q, a, \varepsilon) = \phi$ for all $a \in \Sigma\varepsilon$.
3. For any $q \in$ Q and $a \in \Sigma$, if $(q, a, \varepsilon) \neq \phi$, then $\delta(q, a, s) = \phi$ for all $s \in \Gamma$ and $\delta(q, \varepsilon, t) = \phi$ for all $t \in \Gamma_\varepsilon$
4. For any $q \in Q$, if $\delta(q, \varepsilon, \varepsilon) \neq \phi$, then $\delta(q, a, t) = \phi$ for all $a \in \Sigma\varepsilon$ and $t \in \Gamma_\varepsilon$ (except when $a = \varepsilon$, $t = \varepsilon$).

Rule-2 says that if there is a transition from state q that reads character, s from stack but doesn't read other input, other transitions from q, that don't read stack are not allowed and other transitions from q that read s from the stack and read the input are not allowed either.

Rule-3 says that if there is a transition from state q that reads character a, but doesn't read stack, other transitions from q that don't read the input are not allowed and other transitions from q that read 'a' from input and read the stack are not allowed either.

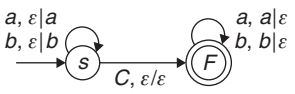
Rule-4 says that if there is a transition from q that doesn't read either input or stack, all other transitions from q are not allowed.

Example 13: A language, L is defined as: $L = \{wcw^R : w \in (a, b)^*\}$. What is Nature of language L?

(A) CFL and DCFL
(B) Only CFL
(C) Only DCFL
(D) None of these

Solution: (A)

$$L = \{x = wcw^R \text{ for } w \in (a, b)^*\}$$



Clearly, obtained PDA is also DPDA in sense; there is no choice in transitions.

$\therefore$ Hence L is CFL and DCFL.

EXERCISES

Practice Problems I

Directions for questions 1 to 15: Select the correct alternative from the given choices.

1. Consider the grammar, $G = (V, \Sigma, R, S)$ where $V = \{a, b, S, A\}$, $\Sigma = \{a, b\}$, $R = \{S \to AA, A \to AAA, A \to a, A \to bA, A \to Ab\}$ How many strings can be generated by $L(G)$ that can be produced by derivations of four or fewer steps?
(A) 5 (B) 10 (C) 14 (D) 8

2. Consider the following languages L_1, L_2 and L_3:
$L_1 = \{a^n b^m c^{n+m} | n, m \geq 0\}$
$L_2 = \{a^n b^{n+1} c^{n+2} | n \geq 0\}$
$L_3 = \{a^n b^n c^m | n, m \geq 0\}$
Which of following statement is true?
(A) L_1, L_2, L_3 are context free languages
(B) L_1, L_2 are context free but not L_3
(C) L_1, L_3 are context free but not L_2
(D) L_1, L_2, L_3 are not context free languages.

3. The language, $L = \{b_i \# b_{i+1} : b_i \text{ is } i \text{ in binary}, i \geq 1\}$ is:
(A) Regular
(B) Context free
(C) Regular and context free
(D) Neither context free nor Regular

4. The CFG, $G : A \to BAB|B|\varepsilon$, $B \to 00|\varepsilon$. The CFG is normalized using CNF. The obtained G', contains ___ rules.
(A) 11 (B) 14 (C) 12 (D) 13

5. The language $L = \{0^{2^i} : i \geq 1\}$ is:
(A) Context free
(B) DCFL
(C) Both CFL and DCFL
(D) Not context free language

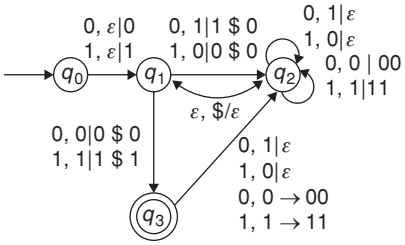
6. The context free grammar, G is defined with production rules $S \to EcC'|aAE|AU$, $A \to aA|\varepsilon$, $B \to bB|\varepsilon$, $C' \to cC'|\varepsilon$, $E \to aEc|F$, $F \to bFc|\varepsilon$, $U \to aUc|V$, $V \to bVc|bB$ What is the language generated by L?
(A) $L = \{a^n b^m c^k : k \neq n + m\}$
(B) $L = \{a^n b^m c^k : k = n + m\}$
(C) $L = \{a^n b^m c^k : k > n + m\}$
(D) $L = \{a^n b^m c^k : k < n + m\}$

7. Consider the grammar, $G \equiv S \to abScB|\varepsilon$, $B \to bB|b$. What language does it generate?
(A) $L(G) = \{(ab)^n (cb)^m | n = m\}$
(B) $L(G) = \{a^n b^n (cb)^m | n \neq m\}$
(C) $L(G) = \{(ab)^n (cb^m)^n | n \geq 0, m > 0\}$
(D) $L(G) = \{\{ab)^n (cb^m)^n | n \geq 0, m \geq 0\}$

8. The language, $L = \{0^i 1^j 2^k | i \neq j \text{ or } j \neq k\}$. The CFG, G generated by L contains ___ rules.
(A) 23 (B) 20
(C) 21 (D) 19

9. The DPDA constructed to accept language, L with property $L = L_1 \cup L_2$ where $L_1 = \{10^n 1^n | n > 0\}$, $L_2 = \{110^n 1^{2n} | n > 0\}$ contains ___ states.
(A) 4 (B) 5
(C) 6 (D) 7

10. The PDA is designed as:



What is the language generated by the above PDA?
(A) Binary strings that have same number of 0's and 1's.
(B) Binary strings that start with 00 and end with 11 and have same number of 0's and 1's.
(C) Binary strings that start and end with the same symbol and have same number of 0's and 1's.
(D) Binary strings that start with 11 and end with 00 and have same number of 0's and 1's.

11. The language, $L = (ba^{m_1} ba^{m_2} b \cdots ba^{m_n} : n \geq 2, m_1, \ldots m_n \geq 0$ and $m_i \neq m_j$ for some $i, j)$. What is nature of 'L'?
(A) Regular
(B) Context free but not regular
(C) Regular but not context free
(D) Neither context free nor regular

12. Two languages L_1, L_2 are defined as:
$L_1 = \{a^i b^j c^k : i, j, k \geq 0, i = j\}$
$L_2 = \{a^i b^j c^k : i, j, k \geq 0, j = k\}$ which of following statements are true?
(i) $L_1 \cap L_2$ is context free
(ii) $L_1 \cap L_2 = \{a^n b^n c^n | n \geq 0\}$
(iii) L_1, L_2 are context free
(iv) Only L_1 is context free
(A) All are true (B) (i), (ii) are true
(C) (iii), (iv) are true (D) (ii), (iii) are true

13. The language generated by grammar:
$S \to Te|Ue$, $T \to cTd|cT|\varepsilon$, $U \to cUd|Ud|dd$. is
(A) $L = \{c^n d^m e : m \geq n\}$
(B) $L = \{c^n d^m e : m = n\}$
(C) $L = \{c^m d^n e^m : m \geq n + 2\}$
(D) None of these

14. Remove null productions, useless symbols from the following grammar result in:
$S \to ABC$
$A \to aBC$
$B \to C|\varepsilon$
$C \to cd|DCF$
$D \to dD|\varepsilon$

$E \to eFE$
$F \to eC$

(A) $S \to ABC|AC$
$A \to aBC|aC$
$B \to C$
$C \to cd|DCF|CF$
$D \to dD|d$
$F \to eC$

(B) $S \to aBCc$
$A \to aBC$
$B \to cD|dDEF|dEF$
$C \to cD|dDEF|dEF$
$F \to eB$
$D \to dD|d$
$E \to eFE|e$

(C) $S \to aBCBC|aBC$
$B \to cD|dDEF|dEF$
$F \to eB$
$C \to dD|d$
$D \to e$
$F \to CD|dDEF$

(D) None of these

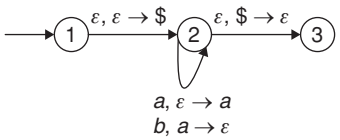
15. Let the language L_1, L_2 are defined as: L_1: $\{a^i\, b^{2i}\, c^j |\ i, j \geq 0\}$, $L_2 = \{a^i\, b^{2i}\, a^i |\ i \geq 0\}$. Which of following is true?
(A) L_1, L_2 are context free
(B) Only L_1 is context free
(C) Only L_2 is context free
(D) Neither L_1 nor L_2 is context free

Practice Problems 2

Directions for questions 1 to 15: Select the correct alternative from the given choices.

1. Consider the alphabet $\Sigma = \{a, b, c, (,), \cup, ^*, \phi\}$. Then context free grammar that generates all strings in Σ^* that are regular expressions over $\{a, b\}$ is:
(A) $S \to S^*|a|b|SS$
(B) $S \to \phi|a|b|S$
(C) $S \to \phi|a \cup b|S^*$
(D) $S \to \phi|S^*|a|b|(S)|S \cup S|SS$

2. The PDA for language, L is designed below. The CFG generated contains ____ productions.



(A) 5 (B) 4
(C) 3 (D) 6

3. The language, L generated by the following grammar, $S \to SS|\ AAA|\varepsilon$, $A \to aA|\ Aa|\ b$ is
(A) $(a^*\, b^*)^*$ (B) $(a^*\, b^*\, b^*\, a^*)^*$
(C) $a^*\, b^*\, a^*$ (D) $(a^*\, b\, a^*\, b\, a^*\, b\, a^*)^*$

4. The grammar, G is defined with rules $S \to S_1|S_2$, $S_1 \to S_1 b|Ab|\varepsilon$, $A \to aAb|\ ab$, $S_2 \to S_2 a|\ Ba|\varepsilon$, $B \to bBa|\ ba$. The CNF is applied on G. The obtained grammar, G' contains ___ rules.
(A) 24 (B) 23
(C) 21 (D) 20

5. The language, $L = \{b^{n^2} :\ n \geq 1\}$ is:
(A) CFL but not DCFL
(B) DCFL but not CFL
(C) Only DCFL
(D) Not CFL

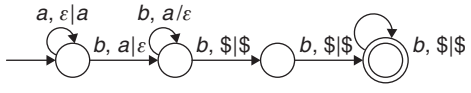
6. Consider the grammar, $G = S \to aSc|B$, $B \to bBc|\varepsilon$ The language, L generated by G is
(A) $L = \{a^n\, b^m\, c^k : k = n + m\}$
(B) $L = \{a^n\, b^m\, c^k : k \neq n + m\}$
(C) $L = \{a^n\, b^m\, c^k : k > n + m\}$
(D) $L = \{a^n\, b^m\, c^k : k < n + m\}$

7. The grammar, G is defined with productions:
$S \to 0A|1B$, $A \to 0AA|1S|1$, $B \to 1BB|0S|0$
The grammar, G_2 is defined with productions:
$S \to AB|aaB$, $A \to a|Aa$, $B \to b$
Which grammar is/are ambiguous?
(A) Only G_1
(B) Only G_2
(C) Both G_1 and G_2
(D) Both G_1 and G_2 are unambiguous

8. The language, $L_1 = \{0^n\, 1^n |\ n > 0\}$ and $L_2 = \{0^n\, 1^{2n} |\ n > 0\}$. The CFG generated for $L_1 \cup L_2$ is:
(A) $S \to 0\ A\ 1|0\ A\ 1\ 1$
$A \to 0|1|\varepsilon$
(B) $S \to 0\ A\ 1\ 1$
$A \to 0|1|\varepsilon$
(C) $S \to 0\ A\ 1|0\ B\ 1\ 1$
$A \to 0\ A\ 1|\varepsilon$
$B \to 0\ B\ 1\ 1|\varepsilon$
(D) $S \to 0\ A\ 1\ 1|0\ 1\ 1$
$A \to 0|1|\varepsilon$

9. The NPDA constructed to accept language, L with property, $L = L_1 \cup L_2$, where $L_1 = \{1^n\, 0^n |\ n > 0\}$, $L_2 = \{0^n\, 1^{2n} |\ n \geq 0\}$ contains ____ final states.
(A) 3 (B) 1
(C) 2 (D) 4

10. The DPDA for language, L is designed below. What is the language generated?



(A) $L = \{a^n b^m : m = n\}$
(B) $L = \{a^n b^m : m = n + 2\}$
(C) $L = \{a^n b^m : m \geq n + 2\}$
(D) $L = \{a^n b^m : m \leq n + 2\}$

11. The CFG, G is defined with rules:
$S \to AB|CD, A \to A00|\varepsilon, B \to B11|1, C \to C00|0, D \to D11|\varepsilon$. The language generated by G is
(A) $L = \{0^n 1^n | n \geq 0\}$
(B) $L = \{0\ 0^n\ 1\ 1^n | n > 0\}$
(C) $L = \{0^n 1^m | n + m \text{ is odd}\}$
(D) $L = \{0^n 1^m | n + m \text{ is even}\}$

12. The languages, L_1, L_2, L_3 are defined as:
$L_1 = \{a^n b^m c^{n+m} | n, m \geq 0\}$, $L_2 = \{a^n b^n c^m | n, m \geq 0\}$, $L_3 = \{a^n b^n c^{2n} | n \geq 0\}$, Which of the following statements are true?
(i) L_1, L_2 are context free
(ii) L_1, L_3 are context free
(iii) $L_3 = L_1 \cap L_2$
(iv) L_1, L_3 are context free but not L_2
(A) (i), (ii) (B) (i), (iii)
(C) (ii), (iii) (D) (iii), (iv)

13. The language, L_1 and L_2 are defined as $L_1 = \{a^n b^n : n \geq 0$ and n is not a multiple of 5$\}$ and $L_2 = \{0^n \# 0^{2n} \# 0^{3n} | n \geq 0\}$. Which of following is true?
(A) L_1 and L_2 are context free
(B) Only L_1 is context free
(C) Only L_2 is context free
(D) Neither L_1 nor L_2 is context free

14. The language, L_1 and L_2 are defined as $\overline{L_1} = \{0^n 1^n)^m | m, n > 0\}$, $L_2 = \{0^n\ 1^n\ 0^n\ 1^n | n \geq 0\}$ which of following is true?
(A) L_1 and L_2 are context free
(B) Only $\overline{L_1}$ is context free
(C) Only L_2 is context free
(D) Neither L_1 nor L_2 is context free

15. The language L_1, L_2 are defined as $L_1 = \{0^i\ 1^i\ 0^j\ 1^i | i, j > 0\}$, $L_2 = \{1^k\ 0^i\ 1^i\ 0^j\ 1^j\ 0^k | i, j, k > 0\}$. Which of following is true?
(A) L_1 and L_2 are context free
(B) Only L_1 is context free
(C) Only L_2 is context free
(D) Neither L_1 nor L_2 is context free

Previous Years' Questions

1. Match the following: **[2008]**

E. Checking that identifiers are declared before their use	P. $L = \{a^n b^m c^n d^m \| n \geq 1, m \geq 1\}$
F. Number of formal parameters in the declaration of a function agrees with the number of actual parameters in use of that function	Q. $X \to XbX \| XcX \| dXf \| g$
G. Arithmetic expressions with matched pairs of parentheses	R. $L = \{wcw \| w \in (a\|b)^*\}$
H. Palindromes	S. $X \to bXb \| cXc \| \varepsilon$

(A) E − P, F − R, G − Q, H − S
(B) E − R, F − P, G − S, H − Q
(C) E − R, F − P, G − Q, H − S
(D) E − P, F − R, G − S, H − Q

2. Consider the languages L_1, L_2 and L_3 as given below.
$L_1 = \{0^p\ 1^q | p, q \in N\}$,
$L_2 = \{\ 0^p\ 1^q | p, q \in N$ and $p = q\}$ and
$L_3 = \{0^p\ 1^q\ 0^r | p, q, r \in N$ and $p = q = r\}$. Which of the following statements is NOT TRUE? **[2011]**
(A) Push Down Automata (PDA) can be used to recognize L_1 and L_2.
(B) L_1 is a regular language.
(C) All the three languages are context free
(D) Turing machines can be used to recognize all the languages.

3. Which of the following problems are decidable? **[2012]**
(1) Does a given program ever produce an output?
(2) If L is a context free language, then, is $\overline{L}$ also context free?
(3) If L is a regular language, then, is $\overline{L}$ also regular?
(4) If L is recursive language, then, is $\overline{L}$ also recursive?
(A) 1, 2, 3, 4 (B) 1, 2
(C) 2, 3, 4 (D) 3, 4

4. Consider the following languages.
$L_1 = \{0^p\ 1^q\ 0^r | p, q, r \geq 0\}$
$L_2 = \{0^p\ 1^q\ 0^r | p, q, r \geq 0, p \neq r\}$
Which one of the following statements is **FALSE**? **[2013]**
(A) L_2 is context-free
(B) $L_1 \cap L_2$ is context-free
(C) Complement of L_2 is recursive
(D) Complement of L_1 is context-free but not regular

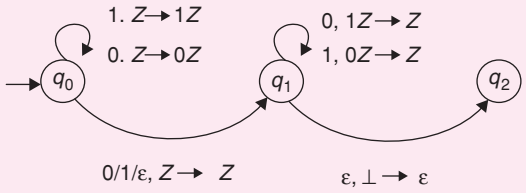
5. Which one of the following is **TRUE**? **[2014]**
(A) The language $L = \{a^n b^n | n \geq 0\}$ is regular
(B) The language $L = \{a^n | n$ is prime$\}$ is regular
(C) The language $L = \{w | w$ has $3k + 1b$'s for some k $\in N$ with $\Sigma = \{a, b\}\}$ is regular
(D) The language $L = \{ww | w \in \Sigma^*$ with $\Sigma = \{0, 1\}\}$ is regular.

6. Consider the following languages over the alphabet $\Sigma = \{0, 1, c\}$.
$L_1 = \{0^n\ 1^n | n \geq 0\}$
$L_2 = \{wcw^r | w \in \{0, 1\}^*\}$
$L_3 = \{ww^r | w \in \{0, 1\}^*\}$

Here w^r is reverse of the string w. Which of these languages are deterministic context-free languages? **[2014]**

(A) None of the languages
(B) Only L_1
(C) Only L_1 and L_2
(D) All the three languages

7. Consider the NPDA $<Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \perp\}, \delta, q_0, \perp, F = \{q_2\}>$, where (as per usual convention) Q is the set of states, Σ is the input alphabet, Γ is the stack alphabet, δ is the state transition function, q_0 is the initial state, $\perp$ is the initial stack symbol, and F is the set of accepting states. The state transition is as follows:



Which one of the following sequences must follow the string 1011 00 so that the overall string is accepted by the automation? **[2015]**

(A) 10110 (B) 10010
(C) 01010 (D) 01001

8. Which of the following languages are context-free? **[2015]**

$L_1 = \{a^m b^n a^n b^m \mid m, n \geq 1\}$
$L_2 = \{a^m b^n a^m b^n \mid m, n \geq 1\}$
$L_3 = \{a^m b^n \mid m = 2n + 1\}$

(A) L_1 and L_2 only (B) L_1 and L_3 only
(C) L_2 and L_3 only (D) L_3 only

9. Consider the following context-free grammars:

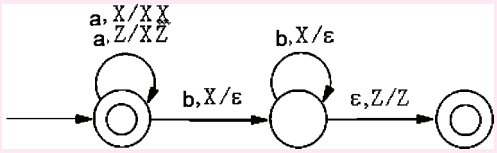
$G_1: S \to aS|B, B \to b|bB$

$G_2: S \to aA|bB, A \to aA|B|\ \varepsilon, B \to bB\varepsilon$

Which one of the following pairs of languages is generated by G_1 and G_2, respectively? **[2016]**

(A) $\{a^m b^n \mid m > 0$ or $n > 0\}$ and $\{a^m b^n \mid m > 0$ and $n > 0\}$
(B) $\{a^m b^n \mid m > 0$and $n > 0\}$ and $\{a^m b^n \mid m > 0$ or $n \geq 0\}$
(C) $\{a^m b^n \mid m \geq 0$ or $n > 0\}$ and $\{a^m b^n \mid m > 0$ and $n > 0\}$
(D) $\{a^m b^n \mid m \geq 0$ and $n > 0\}$ and $\{a^m b^n \mid m > 0$ or $n > 0\}$

10. Consider the transition diagram of a PDA given below with input alphabet $\sum = \{a, b\}$ and stack alphabet = $\{X,Z\}$. Z is the initial stack symbol. Let L denote the language accepted by the PDA.



Which one of the following is TRUE? **[2016]**

(A) $L = \{a^n b^n \mid n \geq 0\}$ and is not accepted by any finite automata.
(B) $L = \{a^n \mid n \geq 0) \cup \{a^n b^n \mid n \geq 0\}$ and is not accepted by any deterministic PDA.
(C) L is not accepted by any Turing machine that halts on every input.
(D) $L = \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$ and is deterministic context-free.

11. Consider the following languages:

$L_1 = \{a^n b^m c^{n+m} : m, n \geq 1\}$
$L_2 = \{a^n b^n c^{2n} : n \geq 1\}$

Which one of the following is TRUE? **[2016]**

(A) Both L_1 and L_2 are context - free.
(B) L_1 is context - free while L_2 is not context - free
(C) L_2 is context - free while L_1 is not context - free.
(D) Neither L_1 nor L_2 is context - free.

12. Consider the following context-free grammar over the alphabet $\Sigma = \{a, b, c\}$ with S as the start symbol:
$S \to abScT \mid abcT$
$T \to bT \mid b$

Which one of the following represents the language generated by the above grammar? **[2017]**

(A) $\{(ab)^n (cb)^n \mid n \geq 1\}$
(B) $\{(ab)^n cb^{m_1} cb^{m_2} \ldots cb^{m_n} \mid n, m_1, m_2, \ldots, m_n \geq 1\}$
(C) $\{(ab)^n (cb^m)^n \mid m, n \geq 1\}$
(D) $\{(ab)^n (cb^n)^m \mid m, n \geq 1\}$

13. If G is a grammar with productions

$$S \to SaS \mid aSb \mid bSa \mid SS \mid \in$$

Where S is the start variable, then which one of the following strings is not generated by G? **[2017]**

(A) *abab* (B) *aaab*
(C) *abbaa* (D) *babba*

14. Consider the context-free rammers over the alphabet $\{a, b, c\}$ given below. S and T are non-terminals.

$$G_1 : S \to aSb|T, T \to cT|\in$$
$$G_2 : S \to bSa|T, T \to cT|\in$$

The language $L(G_1) \cap L(G_2)$ is **[2017]**

(A) Finite
(B) Not finite but regular
(C) Context-Free but not regular
(D) Recursive but not context-free.

15. Consider the following languages over the alphabet $\Sigma = \{a, b, c\}$.

Let $L_1 = \{a^n b^n c^m \mid m, n \geq 0\}$ and $L_2 = \{a^m b^n c^n \mid m, n \geq 0\}$. Which of the following are context-free languages? **[2017]**

I. $L_1 \cup L_2$
II. $L_1 \cap L_2$

(A) I only (B) II only
(C) I and II (D) Neither I nor II

16. Let L_1 L_2 be any two context-free languages and R be any regular language. Then which of the following is/are CORRECT? **[2017]**

I. $L_1 \cup L_2$ is context-free.
II. $\bar{L}_1$ is context-free.
III. $L_1 - R$ is context-free.
IV. $L_1 \cap L_2$ is context-free.

(A) I, II and IV only (B) I and III only
(C) II and IV only (D) I only

17. Identify the language generated by the following grammar, where S is the start variable. **[2017]**

$$S \rightarrow XY$$
$$X \rightarrow aX \mid a$$
$$Y \rightarrow aYb \mid \in$$

(A) $\{a^m b^n \mid m \geq n, n > 0\}$ (B) $\{a^m b^n \mid m \geq n, n \geq 0\}$
(C) $\{a^m b^n \mid m > n, n \geq 0\}$ (D) $\{a^m b^n \mid m > n, n > 0\}$

18. Consider the following languages.

$L_t = \{a^p \mid p \text{ is a prime number}\}$
$L_2 = \{a^n b^m c^{2m} \mid n \geq 0, m \geq 0\}$
$L_3 = \{a^n b^n c^{2n} \mid n \geq 0\}$
$L_4 = \{a^n b^n \mid n \geq 1\}$

Winch of the following are CORRECT? **[2017]**

I. L_1 is context-free but not regular.
II. $L_2$is not context-free.
III. $L_3$is not context-free but recursive.
IV. L_4 is deterministic context-free.

(A) I, II and IV only (B) II and III only
(C) I and IV only (D) III and IV only

19. Consider the following languages:

I. $\{a^m b^n c^p d^q \mid m + p = n + q$, where $m, n, p, q \geq 0\}$
II. $\{a^m b^n c^p d^q \mid m = n$ and $p = q$, where $m, n, p, q \geq 0\}$
III. $\{a^m b^n c^p d^q \mid m = n = p$ and $p \neq q$, where $m, n, p, q \geq 0\}$
IV. $\{a^m b^n c^p d^q \mid mn = p + q$, where $m, n, p, q \geq 0\}$

Which of the languages above are context-free? **[2018]**

(A) I and IV only (B) I and II only
(C) II and III only (D) II and IV only

Answer Keys

Exercises

Practice Problems 1

1. D	**2.** C	**3.** D	**4.** B	**5.** D	**6.** A	**7.** C	**8.** B	**9.** D	**10.** C
11. B	**12.** D	**13.** D	**14.** A	**15.** B					

Practice Problems 2

1. D	**2.** C	**3.** D	**4.** A	**5.** D	**6.** A	**7.** C	**8.** C	**9.** C	**10.** C
11. C	**12.** B	**13.** B	**14.** B	**15.** C					

Previous Years' Questions

1. C	**2.** C	**3.** D	**4.** D	**5.** C	**6.** C	**7.** B	**8.** B	**9.** D	**10.** D
11. B	**12.** B	**13.** D	**14.** B	**15.** A	**16.** B	**17.** C	**18.** D	**19.** B	