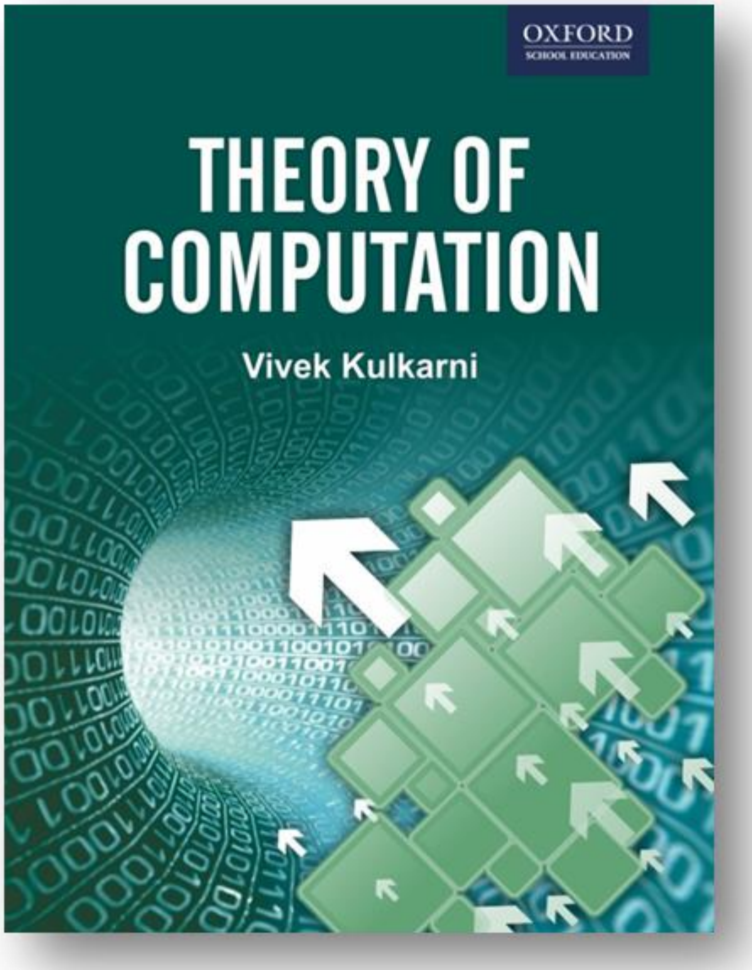


Author: Vivek Kulkarni
(vivek_kulkarni@yahoo.com)

Chapter-5: Grammars

Solutions for Review Questions

Q.1 Write a context-free grammar (CFG) which generates the language L denoted by:

$(a + b)^* \, bbb \, (a + b)^*$

Solution:

The CFG required is,

$\mathrm{S} \to \mathrm{A\ B\ A}$

$\mathrm{A} \to \mathrm{a\ A} \mid \mathrm{b\ A} \mid \in$

$\mathrm{B} \to \mathrm{bbb}$

Q.2 Distinguish between the type-0 and type-1 grammars.

Solution:

Refer to the sections 5.11.1 and 5.11.2.

Q.3 Explain Greibach normal form with the help of an example.

Solution:

Refer to the sections 5.10.2.

Q.4 Consider the grammar $G = \{(S, A), (0, 1), P, S\}$, where P consists of:

$S \to 0\, A\, S \mid 0$

$A \to S\, 1\, A \mid S\, S \mid 1\, 0$

Show the leftmost derivation and rightmost derivation for the input string ‘001100’.

Solution:

Leftmost Derivation:

$S \to 0\, \underline{A}\, S$

$S \to 0\, \underline{S}\, 1A\, S$

$S \to 0\, 0\, 1\underline{A}\, S$

$S \to 0\, 0\, 1\, 1\, 0\, \underline{S}$

$S \to 0\, 0\, 1\, 1\, 0\, 0$

Rightmost Derivation:

$S \rightarrow 0\ A\ \underline{S}$

$S \rightarrow 0\ \underline{A}\ 0$

$S \rightarrow 0\ S\ 1\ \underline{A}\ 0$

$S \rightarrow 0\ \underline{S}\ 1\ 1\ 0\ 0$

$S \rightarrow 0\ 0\ 1\ 1\ 0\ 0$

Q.5 What is a derivation tree? Give suitable examples.

Solution:

Refer to the section 5.6.

Q.6 Convert the following grammar to CNF:

$S \rightarrow A\ B\ A$

$A \rightarrow a\ A \mid \in$

$B \rightarrow b\ B \mid \in$

Solution:

Refer to the example 5.42 from the book.

Q.7 Find the CFL associated with the following CFG:

$S \rightarrow a\ B \mid b\ A$

$A \rightarrow a \mid a\ S \mid b\ A\ A$

$B \rightarrow b \mid b\ S \mid a\ B\ B$

Solution:

Refer to the example 5.6 from the book.

Q.8 Is the following grammar ambiguous?

$S \rightarrow 0\ A\ 1 \mid 0\ B\ A$

$A \rightarrow S\ 0\ 1 \mid 0$

$B \rightarrow 1\ B \mid 1$

Solution:

The grammar is unambiguous, as for any string there is only one way of derivation possible.

Q.9 Show that the following CFG is ambiguous. Remove the ambiguity and write an equivalent unambiguous CFG.

$S \to S + S \mid S * S \mid 4$

Solution:

The grammar is ambiguous. We can show two different ways to derive the string ‘4 + 4 * 4’. Below are the two different leftmost derivations for the same string,

1. $S \to \underline{S} + S$

   $S \to 4 + \underline{S}$

   $S \to 4 + \underline{S} * S$

   $S \to 4 + 4 * \underline{S}$

   $S \to 4 + 4 * 4$

2. $S \to \underline{S} * S$

   $S \to \underline{S} + S * S$

   $S \to 4 + \underline{S} * S$

   $S \to 4 + 4 * \underline{S}$

   $S \to 4 + 4 * 4$

The equivalent CFG after removing the ambiguity is,

$S \to S + T \mid T$

$T \to T * U \mid U$

$U \to 4$

Q.10 Define the following and give appropriate examples:

(a) Derivation graph

(b) Unrestricted grammar

(c) CFG

Solution:

(a) Derivation graph: Refer to the section 5.17.

(b) Unrestricted grammar: Refer to the section 5.11.1.

(c) CFG: Refer to the section 5.11.3.

Q.11 What is regular grammar? State its category in Chomsky hierarchy.

Solution:

: Refer to the section 5.11.4.

Q.12 Write a CFG, which accepts the language L, where:

$L = \{0^i\, 1^j\, 0^k \mid j > i + k\}$.

Using this grammar, show the derivation of the string ‘0111100’.

Solution:

The CFG can be written as,

$S \rightarrow A\ C\ B$

$A \rightarrow 0\ A\ 1 \mid \in$

$B \rightarrow 1\ B\ 0 \mid \in$

$C \rightarrow 1\ C \mid 1$

We can see that A generates equal number of 0’s and 1’s. The symbol B adds equal number of 1’s as it adds the number of 0’s. If we consider the sequence AB, then it generates number of 1’s, $j = i + k$. The symbol C is introduced will generate one or more 1’s to achieve $j > i + k$.

Let us derive the string ‘0111100’ using the above CFG.

$S \Rightarrow \underline{A}\ C\ B$

$\Rightarrow 0\ \underline{A}\ 1\ C\ B$

$\Rightarrow 0\ 1\ \underline{C}\ B$ (using $A \rightarrow \in$)

$\Rightarrow 0\ 1\ 1\ \underline{B}$

$\Rightarrow 0\ 1\ 1\ 1\ \underline{B}\ 0$

$\Rightarrow 0\ 1\ 1\ 1\ 1\ \underline{B}\ 0\ 0$

$\Rightarrow 0\ 1\ 1\ 1\ 1\ 0\ 0$ (using $B \rightarrow \in$)

Q.13 Convert the following right-linear grammar to its equivalent left-linear grammar:

$S \to a\,A \mid b\,B \mid a\,S \mid a$

$A \to b\,A \mid \in$

$B \to a\,B \mid \in$

Solution:

As there is a self-loop on the start symbol '$S \to$ a S', let us introduce a new start symbol T. The modified CFG is,

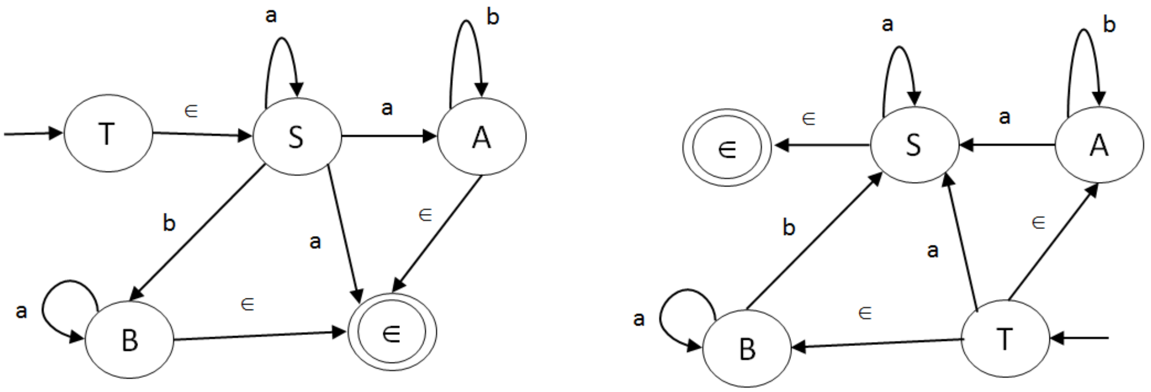
$\mathrm{T} \to \mathrm{S}$

$S \to a\,A \mid b\,B \mid a\,S \mid a$

$A \to b\,A \mid \in$

$B \to a\,B \mid \in$

Let us draw the FA corresponding to the given right-liner grammar. The positions of the start and the final states need to be exchanged followed by the reversal of all the transitions. We get the reversed FA as shown in the figure below.



From the reversed FA we can write the equivalent left-linear grammar as below,

$T \to S\,a \mid B \mid A$

$S \to S\,a \mid \in$

$A \to A\,b \mid S\,a$

$B \to B\,a \mid B\,b$

Q.14 Write the CFG for each of the following languages:

(1) The set of palindromes over alphabet {a, b}

(2) The set of all strings over alphabet $\{a, b\}$ with exactly twice many a's as b's

(3) $L = \{a^i\, b^j\, c^k \mid i \neq j \text{ or } j \neq k\}$

Solution:

(1) $S \rightarrow a\, S\, a \mid b\, S\, b \mid a \mid b \mid \in$

(2) Refer to the example 5.25 from the book.

(3) i ≠ j means, either i > j or, i < j. Same holds true for j ≠ k.

For the condition i ≠ j we can write the CFG as,

$X \rightarrow S\, C$

$S \rightarrow a\, S\, b \mid A \mid B$

$A \rightarrow a\, A \mid \in$

$B \rightarrow b\, B \mid \in$

$C \rightarrow c\, C \mid \in$

Similarly for the condition j ≠ k we can add the following productions,

$X \rightarrow A\, T$

$T \rightarrow b\, T\, c \mid B \mid C$

Therefore, the resultant grammar is,

$X \rightarrow S\, C \mid A\, T$

$S \rightarrow a\, S\, b \mid A \mid B$

$T \rightarrow b\, T\, c \mid B \mid C$

$A \rightarrow a\, A \mid \in$

$B \rightarrow b\, B \mid \in$

$C \rightarrow c\, C \mid \in$

Q.15 Consider the following languages:

$L_1 = \{a^n\, b^{2n}\, c^m \mid n, m \geq 0\}$

$L_2 = \{a^n\, b^m\, c^{2m} \mid n, m \geq 0\}$

Write the CFG for each of them.

Solution:

The CFG for the L_1 is,

$X \rightarrow S\ C$

$S \rightarrow a\ S\ b\ b \mid \in$

$C \rightarrow c\ C \mid \in$

The CFG for the L_2 is,

$X \rightarrow A\ T$

$T \rightarrow b\ T\ c\ c \mid \in$

$A \rightarrow a\ A \mid \in$

Q.16 Write a short note on the applications of CFG.

Solution:

Refer to the section 5.18.

Q.17 Write a context-free grammar (CFG), which defines a language *L* over $\sum = \{a, b\}$ such that *L* contains the words in which, the letter *b* does not appear consecutively three times.

Solution:

The regular expression for such a language can be written as,

$(a^* \text{ b } a^* \text{ b } a^+)^*$

It can be written as CFG as shown below,

$\text{S} \rightarrow X\ S \mid \in$

$X \rightarrow A\ b\ A\ b\ B$

$A \rightarrow a\ A \mid \in$

$B \rightarrow a\ A \mid a$

Q.18 Remove the unit productions from the following CFG.

$S \rightarrow a\ X \mid Y\ b$

$X \rightarrow S$

$Y \rightarrow b\ Y \mid b$

Solution:

Substituting for *X* in *S* we get,

$S \rightarrow a\ S \mid Y\ b$

$Y \rightarrow b\ Y \mid b$

Q.19 Consider grammar *G* defined using the productions:

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

Is the grammar *G* ambiguous? Justify your answer.

Solution:

Refer to the section 5.8.

Q.20 Write a CFG that will generate the language:

$L = \{WCW^R \mid W \in (a, b)^*$, and W^R is the reverse of $W\}$

Using this grammar, show the leftmost derivation, rightmost derivation, and derivation tree for the input string '*ababCbaba*'.

Solution:

The CFG can be written as,

$S \rightarrow a\ S\ a \mid b\ S\ b \mid C$

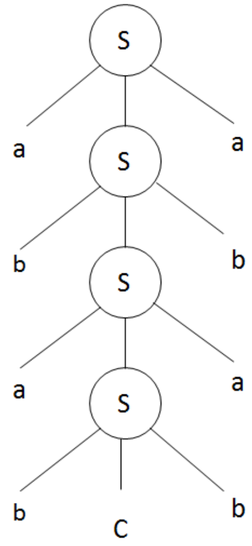
This is nothing but a CFG for all the palindrome strings with middle symbol as 'C'.

Leftmost derivation for '*ababCbaba*':

$$\begin{aligned} S &\Rightarrow a\ \underline{S}\ a \\ &\Rightarrow a\ b\ \underline{S}\ b\ a \\ &\Rightarrow a\ b\ a\ \underline{S}\ a\ b\ a \\ &\Rightarrow a\ b\ a\ b\ \underline{S}\ b\ a\ b\ a \\ &\Rightarrow a\ b\ a\ b\ C\ b\ a\ b\ a \end{aligned}$$

As S is the only non-terminal on the right hand side of any production, even the rightmost derivation looks exactly same.

The derivation tree is as shown below,



Q.21 Write the CFGs generating the following languages, for *n* > 0:

(1) $(bdb)^n C^n$

(2) $(ab)^n\ (cd)^n$

Solution:

(1) S $\rightarrow b\ d\ b\ S\ C \mid b\ d\ b\ C$

(2) S $\rightarrow a\ b\ S\ c\ d \mid a\ b\ c\ d$

Q.22 Find the CFG without $\in$-productions equivalent to the following grammar. Assume *S* is the start symbol.

$S \rightarrow A\ B\ a\ C$

$A \rightarrow B\ C$

$B \rightarrow b \mid \in$

$C \rightarrow D \mid \in$

$D \rightarrow C$

Solution:

Here, *C*, *B*, *D* and A are the nullable non-terminals.

Hence, the CFG without $\in$-productions equivalent to the above grammar can be written as,

$S \rightarrow A\ B\ a\ C \mid A\ B\ a \mid A\ a\ C \mid B\ a\ C \mid A\ a \mid B\ a \mid a\ C \mid a$

$A \rightarrow B\ C \mid B \mid C$

$B \rightarrow b$

$C \rightarrow D$

$D \rightarrow C$

Q.23 Is the language $L = \{a^n\ b^m \mid n \neq m\}$ context-free? If yes, write the CFG defining this language.

Solution:

$S \rightarrow a\ S\ b \mid a\ A \mid b\ B$

$A \rightarrow a\ A \mid \in$

$B \rightarrow b\ B \mid \in$

As $n \neq m$, n can either be greater than m or less than m.

'$S \rightarrow a\ S\ b$' always generates equal number of *a*'s and *b*'s. Once that is done, the middle S can either be derived as *'$S \rightarrow a\ A$' or '$S \rightarrow b\ B$'* which generates more number of *a*'s than *b*'s which is, $n > m$ or more number of *b*'s than *a*'s which is, $n < m$.

Q.24 Convert the following grammar into CNF:

$S \rightarrow a\ B \mid b\ A$

$A \rightarrow a \mid a\ S \mid b\ A\ A$

$B \rightarrow b \mid b\ S \mid a\ B\ B$

Solution:

Refer to the example 5.41 from the book.

Q.25 Write the CFG for the language:

$L = \{a^{2n}\ b^n \mid n \geq 1\}$

Solution:

The CFG can be written as,

$S \rightarrow$ a a S b | a a b

Q.26 What is an ambiguous grammar? Explain with the help of an example, the removal of ambiguity in CFGs.

Solution:

Refer to the section 5.8.

Q.27 Convert the following CFG into CNF:

$S \rightarrow\ a\,S\,a \mid b\,S\,b \mid a \mid b \mid a\,a \mid b\,b$

Solution:

Refer to the example 5.40 from the book.

Q.28 Write a short note on Chomsky hierarchy.

Solution:

Refer to the section 5.11.

Q.29 Find the context-free grammar with no useless symbols equivalent to:

$S \rightarrow A\,B \mid C\,a$

$B \rightarrow B\,C \mid A\,B$

$A \rightarrow a$

$C \rightarrow a\,B \mid b$

Solution:

As we can see, symbol *B* does not always derive string of all terminals. Look at the production rules for *B*, that are, $B \rightarrow B\,C \mid A\,B$. Both *C* and *A* can derive string of terminals. Hence, *B* is a useless symbol. We need to remove symbol *B* and all its usage from the above CFG to get,

$S \rightarrow A\ \mid C\,a$

$A \rightarrow a$

$C \rightarrow b$

Q.30 If the grammar *G* is given by the productions:

$S \to a\,S\,a \mid b\,S\,b \mid a\,a \mid b\,b \mid \in,$

Show that $L(G)$ has no strings of odd length.

Solution:

Let us try deriving strings with their increasing string lengths.

The minimal length, i.e., zero length string is '$\in$'. This is an even length string, zero being the even number.

The next level strings which are directly derivable from start symbol S are, 'aa' and 'bb'. These have length two which is also a even number.

Consider the below derivations,

$S \Rightarrow \text{a } \underline{\text{S}} \text{ a}$	$S \Rightarrow \text{a } \underline{\text{S}} \text{ a}$
$S \Rightarrow \text{a b } \underline{\text{S}} \text{ b a}$	$S \Rightarrow \text{a b } \underline{\text{S}} \text{ b a}$
$S \Rightarrow \text{a b b a}$	$S \Rightarrow \text{a b a a b a}$

It always adds the even length using the remaining productions for S.

Thus, the language $L(G)$ has no strings of odd length.

Q.31 Construct a DFA equivalent to the following grammar:

$S \to a\,S \mid b\,S \mid a\,A$

$A \to b\,B$

$B \to a\,C$

$C \to b\,A \mid \in$

Solution:

As there is a self-loop on the start symbol, let us introduce a new start symbol T. The modified CFG is,

$T \to S$

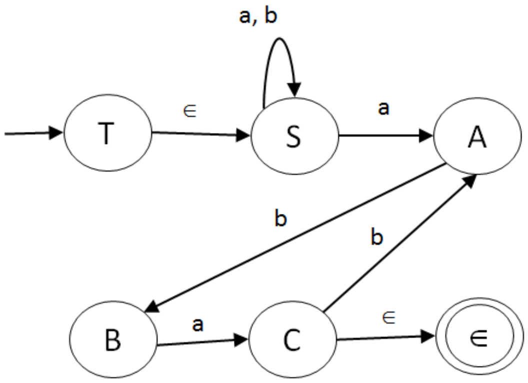
$S \to a\,S \mid b\,S \mid a\,A$

$A \to b\,B$

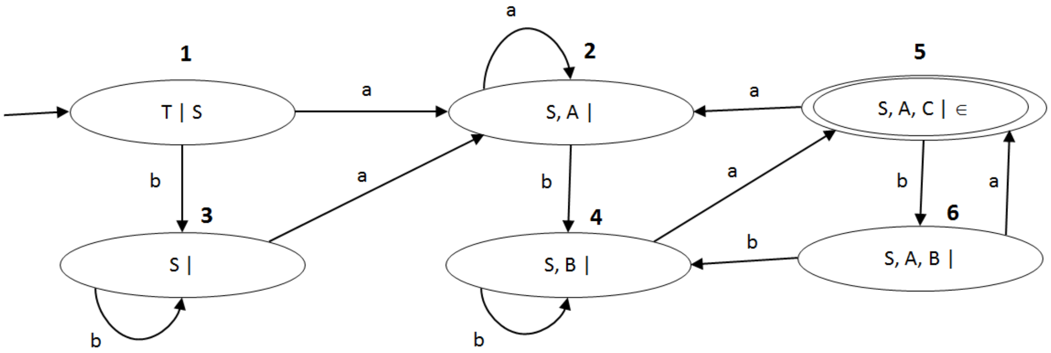
$B \to a\,C$

$C \to b\,A \mid \in$

The NFA corresponding to the above CFG is as shown in the figure below.



This is NFA with ∈-transitions and can be directly converted to the equivalent DFA.



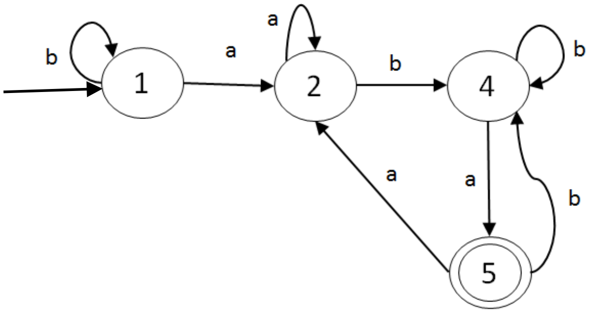
We have relabelled the states from 1 to 6. The STF for the DFA is,

Q \ ∑	a	b
1	2	3
2	2	4
3	2	3
4	5	4
* 5	2	6
6	5	4

We can see that the states (1, 3) and (4, 6) are equivalent. To obtain the reduced DFA we can remove states 3 and 6 and replace all occurrences of them by 1 and 4 respectively. The reduced STF is,

Q \ ∑	a	b
1	2	1
2	2	4
4	5	4
* 5	2	4

This STF cannot be reduced further and hence the STF for the final DFA required. The TG for the DFA is,



Q.32 Construct a regular grammar for the DFA shown in Fig. 5.28.

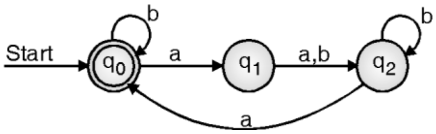


Figure 5.28

Solution:

The grammar required is,

$q0 \rightarrow$ a $q1 \mid b\ q0 \mid \in$

$q1 \rightarrow a\ q2 \mid b\ q2$

$q2 \rightarrow bq2 \mid a\ q0$

Q.33 Convert the following grammar into its equivalent GNF:

$S \rightarrow A\ B;\ A \rightarrow B\ S \mid b;\ B \rightarrow S\ A \mid a$

Solution:

Refer to the example 5.44 from the book; replace S = A_1, A = A_2, B = A_3.

Q.34 Show that the following grammar is ambiguous:

$S \rightarrow a \mid a\ b\ S\ b \mid a\ A\ b$

$A \rightarrow b\ S \mid a\ A\ A\ b$

Solution:

We need to show at least two different derivations for the same string to show that the given CFG is ambiguous. One such string is ‘*abab*’. Below are two different leftmost derivations for the string.

$$\begin{aligned} S &\Rightarrow a\ b\ \underline{S}\ b \\ &\Rightarrow a\ b\ a\ b \end{aligned}$$

$$\begin{aligned} S &\Rightarrow a\ \underline{A}\ b \\ &\Rightarrow a\ b\ S\ b \\ &\Rightarrow a\ b\ a\ b \end{aligned}$$

As there are two different ways to derive the same string, the grammar is ambiguous.

Q.35 Describe the terms ‘left-linear’ and ‘right-linear’ grammars; and convert the following left-linear grammar to its equivalent right-linear grammar.

$S \rightarrow B1 \mid A0 \mid C0$

$A \rightarrow C0 \mid A1 \mid B1 \mid 0$

$B \rightarrow B1 \mid 1$

$C \rightarrow A0$

Solution:

Refer to the section 5.11.4 for details on left-linear and right-liner grammars. Refer to the example 5.48 from the book.

Q.36 For the following grammar, find an equivalent grammar, which is in reduced form and has no unit productions.

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow C \mid b$

$C \rightarrow D$

$D \rightarrow E$

$E \rightarrow a$

Solution:

Substitute for *E* in production for *D* to reduce the unit production '$D \rightarrow E$'.

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow C \mid b$

$C \rightarrow D$

$D \rightarrow a$

Similarly, substituting for D we get,

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow C \mid b$

$C \rightarrow a$

We can get rid of another unit production '$B \rightarrow C$' the similar way,

$S \rightarrow AB$

$A \rightarrow a$

$B \rightarrow a \mid b$

Further simplification gives us,

$S \rightarrow \mathrm{a}\, B$

$B \rightarrow a \mid b$

The final simplified grammar without any unit productions is,

$S \rightarrow a\, a \mid a\, b$

Q.37 Give the regular expression for the language generated by the following grammar:

$S \to A \mid B; A \to 0A \mid \in; \; B \to 0B \mid B \mid \in$

Solution:

We can see that the unit production '$B \to B$' is redundant and can be removed without changing the meaning. We get,

$S \to A \mid B$

$A \to 0A \mid \in$

$B \to 0B \mid \in$

As we can see both A and B generate the regular language 0^*. Hence, the language generated by the grammar where S is the start symbol is,

$0^* + 0^*$

$= 0^*$

Q.38 For the following grammar G, construct an NFA M such that $L(M) = L(G)$:

$S \to a\, b\, A \mid b\, a\, B$

$A \to b\, S \mid b$

$B \to a\, S \mid a$

Solution:

Let us first rewrite the CFG in the form suitable to draw the NFA.

$S \to a\, T \mid b\, U$

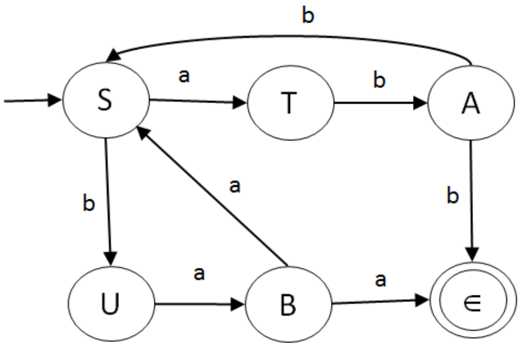
$T \to b\, A$

$U \to a\, B$

$A \to b\, S \mid b$

$B \to a\, S \mid a$

Now, we can draw the NFA equivalent to the above CFG as shown in the figure below.



Q.39 Show that the following language L is a CFL.

$L = \{0^n\, 1^n \mid n \geq 1\} \cup \{0^n\, 1^{2n} \mid n \geq 1\}$

Solution:

$\{0^n\, 1^n \mid n \geq 1\}$ is a CFL as we know and so is $\{0^n\, 1^{2n} \mid n \geq 1\}$. As per closure properties of CFLs, these are closed under union operation and hence, L is a CFL.

Let us try to write CFGs for both the above language parts. The CFG for $\{0^n\, 1^n \mid n \geq 1\}$ is,

$A \rightarrow 0\,A\,1 \mid 0\,1$

The CFG for $\{0^n\, 1^{2n} \mid n \geq 1\}$ is,

$B \rightarrow 0\,B\,1\,1 \mid 0\,1\,1$

Hence, the CFG for L can be written as,

$S \rightarrow A \mid B$

$A \rightarrow 0\,A\,1 \mid 0\,1$

$B \rightarrow 0\,B\,1\,1 \mid 0\,1\,1$

Thus, L is a CFL.

Q.40 Construct the right-linear grammar corresponding to the regular expression:

$R = (0 + 1)^* \,(1 + (01)^*)$

Solution:

$S \rightarrow A\,C$

$A \rightarrow 0\ A \mid 1\ A \mid \in$, represents $(0 + 1)^*$

$B \rightarrow 0\ 1\ B \mid \in$, represents $(01)^*$

$C \rightarrow B \mid 1$, represents $(1 + (01)\ ^*)$

Let us try writing this in the right-linear grammar format.

$S \rightarrow 0\ S \mid 1\ S \mid B \mid 1$

This can be further rewritten as,

$S \rightarrow 0\ S \mid 1\ S \mid 0\ 1\ T \mid 1 \mid \in$

$T \rightarrow 0\ 1\ T \mid \in$

Q.41 Convert the following grammar to Greibach normal form (GNF):

$S \rightarrow B\ S$; $S \rightarrow A\ a$; $A \rightarrow b\ c$; $B \rightarrow A\ c$

Solution:

Let us substitute for *A* and *B* in *S*. We get,

$S \rightarrow A\ c\ S$

$S \rightarrow \mathrm{b\ c\ a}$

$\mathrm{A} \rightarrow \mathrm{b\ c}$

This further can be simplified as,

$S \rightarrow \mathrm{b\ c}\ c\ S$

$S \rightarrow \mathrm{b\ c\ a}$

This can be further updated to rewrite in GNF as below,

$S \rightarrow b\ T$

$S \rightarrow b\ U$

$T \rightarrow c\ V$

$U \rightarrow c\ X$

$\mathrm{V} \rightarrow c\ S$

$X \rightarrow a$

Q.42 Show that the following languages are context-free languages.

(1) $L_1 = \{a^i\, b^i\, c^j \mid i, j >= 1\}$

(2) $L_2 = \{a^i\, b^j\, c^j \mid i, j >= 1\}$

Solution:

Let us try writing the CFGs for these languages.

(1) The CFG for the language, $L_1 = \{a^i\, b^i\, c^j \mid i, j >= 1\}$ is,

$\mathrm{S} \rightarrow A\ C$

$A \rightarrow\ a\ A\ b \mid a\ b$

$C \rightarrow c\ C \mid c$

(2) The CFG for the language, $L_2 = \{a^i\, b^j\, c^j \mid i, j >= 1\}$ is,

$\mathrm{S} \rightarrow A\ C$

$A \rightarrow\ a\ A \mid a$

$C \rightarrow b\ C\ c \mid b\ c$

Q.43 Find whether the string ‘*aabbb*’ is a member of $L(G)$, where G is defined as:

$S \rightarrow XY$

$X \rightarrow YY \mid a$

$Y \rightarrow XY \mid b$

Solution:

Let us try deriving the string ‘aabbb’ using the given grammar. Below is the rightmost derivation for the string.

$S \Rightarrow X\ \underline{Y}$

$\Rightarrow X\ X\ \underline{Y}$

$\Rightarrow X\ \underline{X}\ b$

$\Rightarrow X\ \mathrm{Y}\ \underline{Y}\ b$

$\Rightarrow X\ \underline{Y}\ b\ b$

$\Rightarrow X\ X\ \underline{Y}\ b\ b$

$\Rightarrow X\ \underline{X}\ b\ b\ b$

$$\Rightarrow \quad \underline{X}\, a\, b\, b\, b$$

$$\Rightarrow \quad a\, a\, b\, b\, b$$

Hence, from the above derivation, we can say that 'aabbb' is a member of L(G).

Q.44 Is the following CFG is ambiguous?

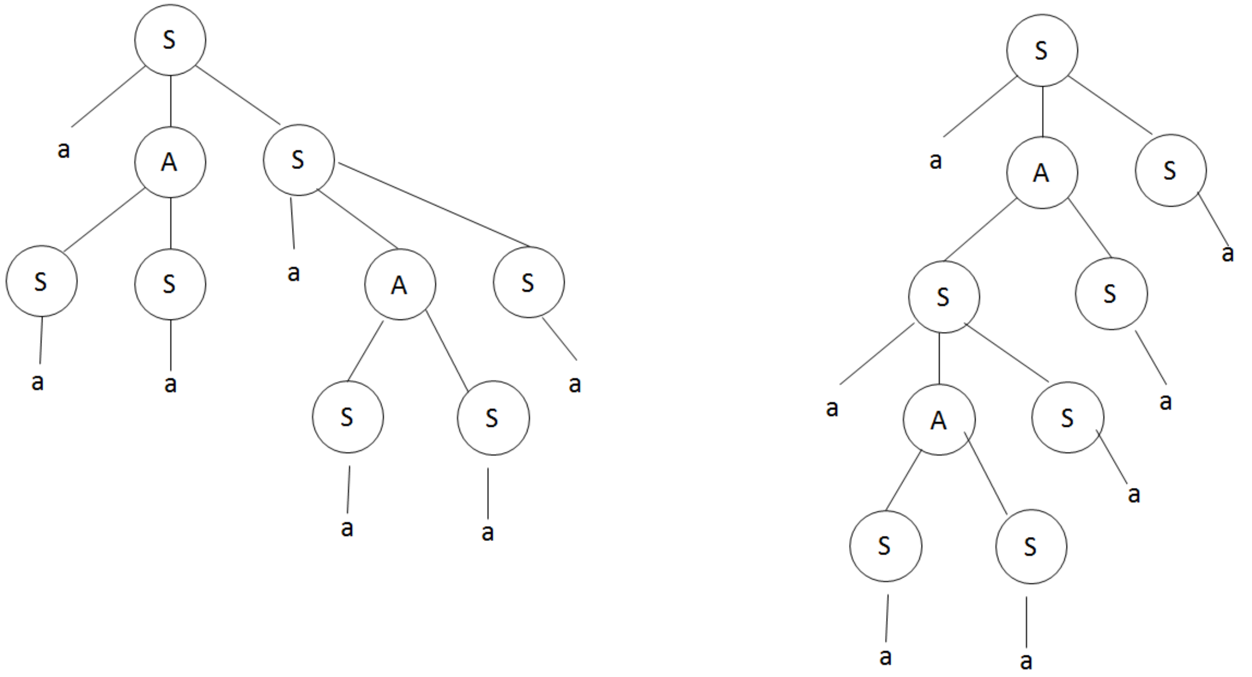
$G = \{(S, A), (\text{a, b}), P, S\}$, where P consists of:

$S \to a\, A\, S \mid a$

$A \to S\, b\, A \mid S\, S \mid b\, a$

Solution:

Let us consider the string ‘aaaaaaa’ that can have at least two different derivations. Hence, the grammar is ambiguous.



Q.45 Convert the following grammar to Greibach normal form, assuming A_1 as the start symbol.

$A_1 \to A_2\, A_3$

$A_2 \to A_3\, A_1 \mid b$

$A_3 \to A_1\, A_2 \mid a$

Solution:

Refer to the example 5.44 from the book.

Q.46 Find context-free grammars generating each of the following languages:

(1) $L_1 = \{a^i\, b^j\, c^k \mid i = j + k\}$

(2) $L_2 = \{a^i\, b^j\, c^k \mid j = i + k\}$

(3) $L_3 = \{a^i\, b^j\, c^k \mid i = j \text{ or } j = k\}$

Solution:

(1) Refer to the example 5.19 from the book.

(2) The CFG for the language, $L_2 = \{a^i\, b^j\, c^k \mid j = i + k\}$ is,

$X \rightarrow S\ A$

$S \rightarrow a\ S\ b \mid \in$

$A \rightarrow b\ A\ c \mid \in$

(3) The CFG for the language, $L_3 = \{a^i\, b^j\, c^k \mid i = j \text{ or } j = k\}$ is,

$P \rightarrow X \mid Y$

(Below 3 production cater to the condition '$i = j$')

$X \rightarrow S\ C$

$S \rightarrow a\ S\ b \mid \in$

$C \rightarrow \mathrm{c\ C} \mid \in$

(Below 3 production cater to the condition '$j = k$')

$Y \rightarrow \mathrm{A\ T}$

$A \rightarrow \mathrm{c\ A} \mid \in$

$T \rightarrow b\ T\ c \mid \in$

Q.47 Draw an NFA that accepts the language generated by the following grammar, and explain the language it generates.

$S \rightarrow a\ A \mid b\ C \mid b$

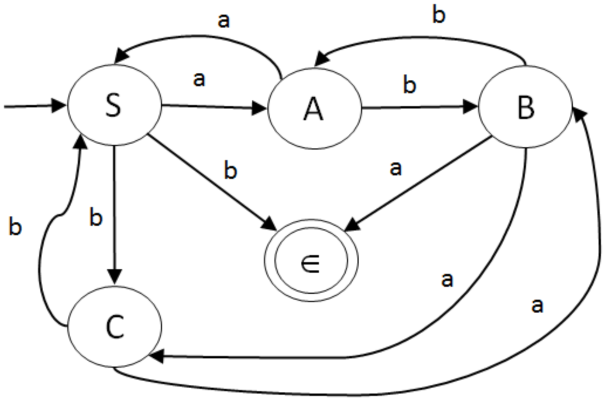
$A \rightarrow a\ S \mid b\ B$

$B \rightarrow a\ C \mid b\ A \mid a$

$C \rightarrow a\ B \mid b\ S$

Solution:

The NFA equivalent to the above grammar is,



The language generated by the given grammar *G* is,

L(G) = {b, aab, baa, aba, bbb, aabaa, …}

Each string in the language is of odd length and contains even number of *a*’s.

Q.48 What is a normal form? Explain Chomsky normal form (CNF) and Greibach normal form (GNF) with the help of suitable examples.

Solution:

Refer to the section 5.10.

Q.49 Find the context-free grammar generating the following languages:

(a) The set of odd length strings in $\{a, b\}^*$ having *a* as the middle symbol

(b) The set of even length strings over $\{a, b\}^*$ whose middle symbols are same

Solution:

(a) The CFG for the ‘set of odd length strings in $\{a, b\}^*$ having *a* as the middle symbol’ is,

$$S \to B\ a\ B$$

$$B \to a\ B \mid b\ B \mid a \mid b$$

(b) The CFG for the ‘set of even length strings over $\{a, b\}^*$ whose middle symbols are same’ is,

$$\text{S} \to B\ a\ a\ B \mid B\ b\ b\ B$$

$$B \to a\ B \mid b\ B \mid a \mid b$$

Q.50 Convert the following CFG into Greibach normal form (GNF):

$S \rightarrow A\,B \mid 0$

$A \rightarrow B\,X \mid 1$

$B \rightarrow C\,D \mid 2$

$C \rightarrow A\,D \mid 0$

$D \rightarrow 1$

Solution:

In the above grammar *X* is a useless symbol. Hence, we need to remove all the production it is part of. Simplifying the grammar without useless symbols is,

$S \rightarrow A\,B \mid 0$

$A \rightarrow 1$

$B \rightarrow C\,D \mid 2$

$C \rightarrow A\,D \mid 0$

$D \rightarrow 1$

Substitute for *A* in the above grammar to get,

$S \rightarrow 1\,B \mid 0$

$B \rightarrow C\,D \mid 2$

$C \rightarrow 1\,D \mid 0$

$D \rightarrow 1$

Substituting *C* in *B* we get,

$S \rightarrow 1\,B \mid 0$

$B \rightarrow 1\,D\,D \mid 0\,D \mid 2$

$C \rightarrow 1\,D \mid 0$

$D \rightarrow 1$

We need not simplify the grammar as it is already in GNF.

Q.51 Construct a CFG for the following sets:

(i) $\{a^{2n}\,bc \mid n \geq 1\}$

(ii) $\{a^n\,b^m\,c^m\,d^n \mid m, n \geq 1\}$

Solution:

(i) The CFG for the language, $\{a^{2n}\, bc \mid n \geq 1\}$ is,

$S \rightarrow A\ b\ c$

$A \rightarrow a\ a\ A \mid a\ a$

(ii) The CFG for the language, $\{a^n\ b^m\ c^m\ d^n \mid m, n\ \geq 1\}$ is,

$S \rightarrow a\ S\ d \mid A$

$A \rightarrow b\ A\ c \mid b\ c$

Q.52 Convert the following CFG to Chomsky normal form:

$S \rightarrow A\ A\ C\ D$

$A \rightarrow a\ A\ b \mid \in$

$C \rightarrow a\ C \mid a$

$D \rightarrow a\ D\ a \mid b\ D\ b \mid \in$

Solution:

CNF has either two non-terminals or a single terminal symbol on the right hand side of every production rule. The above CFG can be written in CNF as below,

$S \rightarrow A\ T$

$T \rightarrow A\ U$

$U \rightarrow C\ D$

$A \rightarrow P\ R \mid \in$

$C \rightarrow P\ C \mid a$

$D \rightarrow P\ V \mid Q\ W \mid \in$

$P \rightarrow a$

$Q \rightarrow b$

$R \rightarrow A\ Q$

$V \rightarrow D\ P$

$W \rightarrow D\ Q$

Q.53 Show that the language $L = \{0^{2^i} \mid i \geq 1\}$ is not a CFL.

Solution:

Step 1: Let us assume that the language L is a CFL.

Step 2: Let us choose a sufficiently large string z such that $z = 0^l$ for some large l. Since we assumed that L is a CFL and an infinite language; pumping lemma can be applied now. This means that we should be able to write a string: $z = uvwxy$.

Step 3: As per pumping lemma, every string 'uv^iwx^iy', for all $i \geq 0$ is in L.

That is, if we consider $i = 0$, then $uv^0wx^0y = uwy$ is in L. Let us assume $|\ uwy\ |$ is a perfect power of 2, say k.

As per pumping lemma, let us have $|\ vx\ | = r \geq 1$; and let $i = k$. Then, as per pumping lemma, 'uv^kwx^ky' should be in L. However, $|\ uv^kwx^ky\ | = |\ uwy\ | + k * |\ vx\ | = k + k * r = k\ (1 + r)$, which is not always a perfect power of 2. Thus, 'uv^kwx^ky' is not in L. This contradicts our assumption that L is a CFL. Hence, L is not a CFL.

Q.54 Using pumping lemma, show that the language $L = \{0^{i^2} |\ i \geq 1\}$ is not a CFL.

Solution:

Step 1: Let us assume that the language L is a CFL.

Step 2: Let us choose a sufficiently large string z such that $z = 0^l$ for some large l, which is a perfect square. Since we assumed that L is a CFL and an infinite language; pumping lemma can be applied now. This means that we should be able to write a string: $z = uvwxy$.

Step 3: As per pumping lemma, every string 'uv^iwx^iy', for all $i \geq 0$ is in L.

That is, if we consider $i = 0$, then $uv^0wx^0y = uwy$ is in L. Let us assume $|\ uwy\ |$ is a perfect square, say k.

As per pumping lemma, let us have $|\ vx\ | = r \geq 1$; and let $i = k$. Then, as per pumping lemma, 'uv^kwx^ky' should be in L. However, $|\ uv^kwx^ky\ | = |\ uwy\ | + k * |\ vx\ | = k + k * r = k\ (1 + r)$, which is not always a perfect square number. Thus, ‘uv^kwx^ky’ is not in L. This contradicts our assumption that L is a CFL. Hence, L is not a CFL.