Problem

Consider the following CFG G:

$$S \rightarrow SS \mid T$$
$$T \rightarrow \mathtt{a}T\mathtt{b} \mid \mathtt{ab}$$

Describe L(G) and show that G is ambiguous. Give an unambiguous grammar H where L(H) = L(G) and sketch a proof that H is unambiguous.

Step-by-step solution

Step 1 of 10

CFG:

• A Context Free Grammar (CFG) is an arrangement of recursive rewriting principles (or productions) used to create pattern of strings.

• A CFG comprises of the following components: An arrangement of terminal symbols.

• Which are the characters of the letter set that show up in the strings created by the grammar.

Comment

Step 2 of 10

Ambiguous Grammar:

• This is a context free grammar to which there exists a string that can have in excess of left most derivation or parse tree.

Comment

Step 3 of 10

The language of a CFG G has to be described and shown to be ambiguous.

An unambiguous grammar H has to obtain from the grammar.

The rules for the CFG G are:

• First, consider the strings produced by the variable T. Let the CFG $I = (V, \Sigma, R, T)$ be:

$T \rightarrow aTb \mid ab$

• As it is either a string of two terminals ab or it is placed between the same two terminals aTb.

• Therefore, the language generated will consist of a sequence of two or more a's followed by the same number of b's.

$L(I) = \{a^i b^i \mid i > 1\}$

• All the strings lying in this language will be of even length as $|a^i b^i| = i + i = 2i$.

Comment

Step 4 of 10

• The start symbol S either can be replaced by two start symbols SS or by the variable T.

• It can be seen that a string in the language $L(G)$ will be the concatenation of one or more occurrences of strings of $L(I)$. So, the language $L(G)$ will be given by:

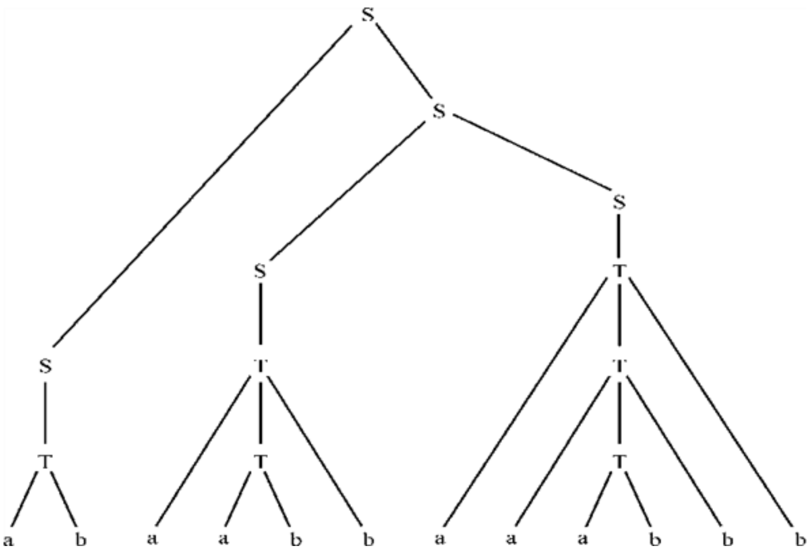
$L(G) = \{a^{i_1}b^{i_1}a^{i_2}b^{i_2}....a^{i_k}b^{i_k} \mid i_1, i_2, ..., i_k \geq 1 \, and \, k \geq 1\}$

• Take the string $s = abaabbaaabbb$.

• A derivation for this string s is:

$$S \Rightarrow SS \Rightarrow TS \Rightarrow abS \Rightarrow abSS \Rightarrow abTS \Rightarrow abaTbS \Rightarrow abaabbS$$
$$\Rightarrow abaabbT \Rightarrow abaabbaTb \Rightarrow abaabbaaTbb \Rightarrow abaabbaaabbb$$

• This leads to parse tree:



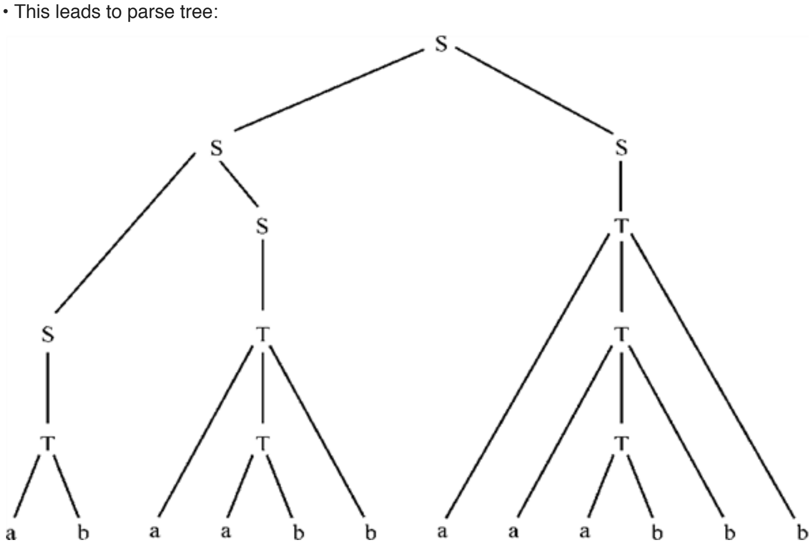
Comment

Step 5 of 10

The Second alternative derivation for this string s is:

$$S \Rightarrow SS \Rightarrow ST \Rightarrow SaTb \Rightarrow SaaTbb \Rightarrow Saaabbb \Rightarrow SSaaabbb \Rightarrow STaaabbb$$
$$\Rightarrow SaTbaaabbb \Rightarrow Saabbaaabbb \Rightarrow Taabbaaabbb \Rightarrow abaabbaaabbb$$

• This leads to parse tree:



• The string $s = abaabbaaabbb$ can be derived into two different parse trees. So, the grammar G is ambiguous. The rules for the start symbol S are:

$S \rightarrow SS \mid T$

• Due to this rule, it is possible to get different derivations for a string in the language $L(G)$.

Comment

Step 6 of 10

Unambiguous Grammar:

• This is also a context free grammar to which each substantial string has a one of a kind unique left most derivation or parse tree.

• The grammar G can be converted into unambiguous grammar $H = (V, \Sigma, R^{'}, S)$ by removing this ambiguity. The rules are modified to:

$S \to TS \mid T$
$T \to aTb \mid ab$

• It has to be checked if $L(G) = L(H)$.

• That is a string s lies in $L(G)$ if and only if it lies in $L(H)$. This result is proven via induction on the length of s.

•

Comment

Step 7 of 10

It has to be proven in both the directions – the 'if' direction and the 'only if' direction.

Comment

Step 8 of 10

Now, it can be proved that a string s lies in $L(G)$ if it lies in $L(H)$.

Basis:

• The string ab lies in $L(H)$.

• It also be derived in $L(G)$, that is $S \Rightarrow_G T \Rightarrow_G ab$

Inductive step:

• All strings of length at most n of $L(H)$ lie in the language $L(G)$. Consider an arbitrary w string of length $n+2$ lying in the language $L(H)$.

• There are three cases possible:

1. The string w starts with an ab. It can be derived in the language $L(G)$ to a string of length n as follows.

$S \Rightarrow_G SS \Rightarrow_G TS \Rightarrow_G abS \Rightarrow_G abx$

2. An ab in the middle of the string w. The derivation for this string is.

$S \Rightarrow_G SS \Rightarrow_G SSS \Rightarrow_G STS \Rightarrow_G SaTbS \Rightarrow_G xaybz$

3. The string w ends with an ab.

$S \Rightarrow_G SS \Rightarrow_G ST \Rightarrow_G Sab \Rightarrow_G xab$

• In all the above cases, the string can be derived in grammar G to a string(s) whose length $\le n$. So, a string s lies in $L(G)$ if it lies in $L(H)$.

Comment

Step 9 of 10

Only-If:

• It can be proved that a string s lies in $L(H)$ if it lies in $L(G)$.

Basis:

• The string ab lies in $L(G)$. It also be derived in $L(H)$, that is $S \Rightarrow_H T \Rightarrow_H ab$.

Inductive step:

• All strings whose length is equal to or less than n of $L(G)$ lie in the language $L(H)$

Comment

Consider an arbitrary w string of length $n+2$ lying in the language $L(G)$.

• The three cases possible are:

• The string w starts with an ab. It can be derived in the language $L(H)$ to a string of length n as follows.

$S \Rightarrow_H SS \Rightarrow_H TS \Rightarrow_H abS \Rightarrow_H abx$

• An ab in the middle of the string w. The derivation for this string is.

$S \Rightarrow_H SS \Rightarrow_H SSS \Rightarrow_H STS \Rightarrow_H SaTbS \Rightarrow_H xaybz$

• The string w ends with an ab.

$S \Rightarrow_H SS \Rightarrow_H ST \Rightarrow_H Sab \Rightarrow_H xab$

The string can be derived in all the cases into a string(s) whose length is at most.

• Therefore, it has been proven if a string lies in $L(H)$ then it also lies in $L(G)$.

• An outline of a proof showing that H **is unambiguous is to be given**.

• A grammar is unambiguous if there is only one leftmost derivation of a string in the grammar.

• An induction on the length of the string can be used to prove this and showing that is only one way to derive an arbitrary string into a string of smaller length.

Hence, it has been proven that the languages $L(H)$ and $L(G)$ are equivalent. In other words:

$L(G) = L(H)$

Comment