Example 6.13 Check whether the grammar is ambiguous or not.

$$S \to aS/AS/A$$
$$A \to AS/a$$

Solution: Consider the string 'aaa'. The string can be generated in many ways. Here, we are giving two ways.

$i)S \to aS \to aAS \to aaS \to aaA \to aaa$

$ii)S \to A \to AS \to ASS \to aAS \to aaS \to aaA \to aaa$

The parse trees for derivation (i) and (ii) are shown in Figs. 6.6 (i) and (ii).

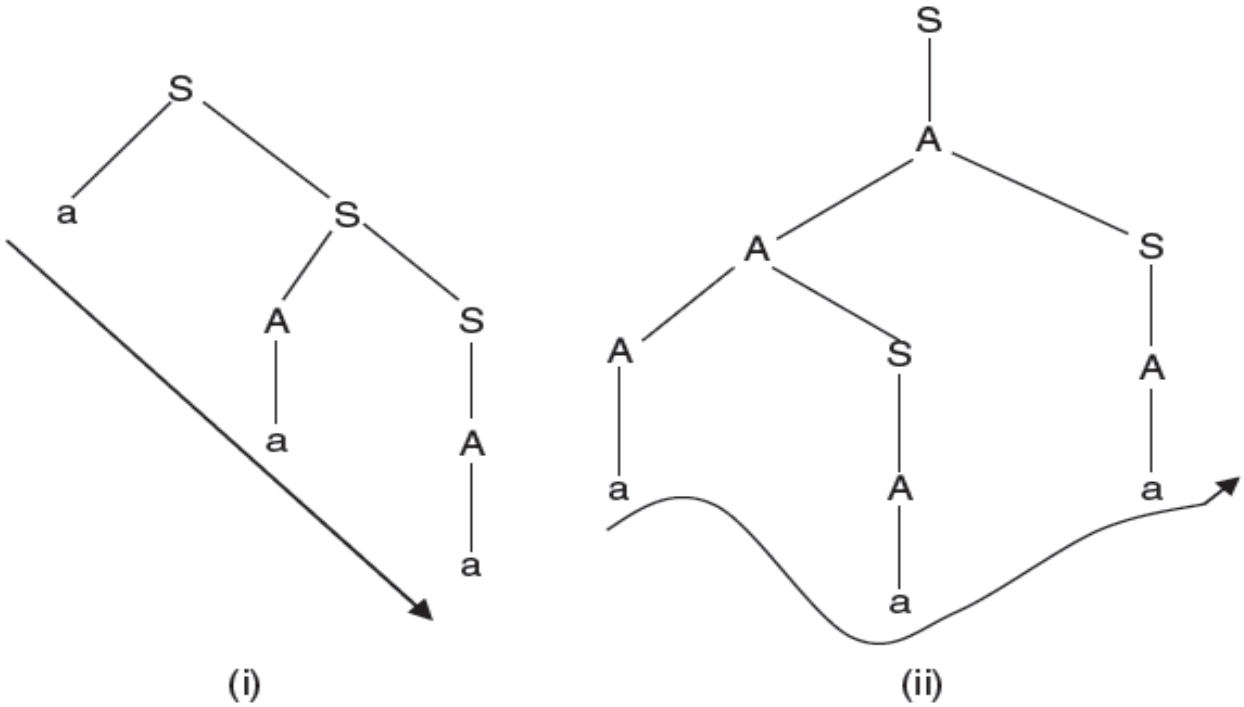


Fig. 6.6

Here, for the same string derived from the same grammar we are getting more than one parse tree. So, according to the definition, the grammar is an ambiguous grammar.

Example 6.14 Prove that the following grammar is ambiguous.

$$V_N : \{S\}$$
$$\Sigma : \{id, +, *\}$$
$$P : S \to S + S / S * S / id$$
$$S : \{S\}$$

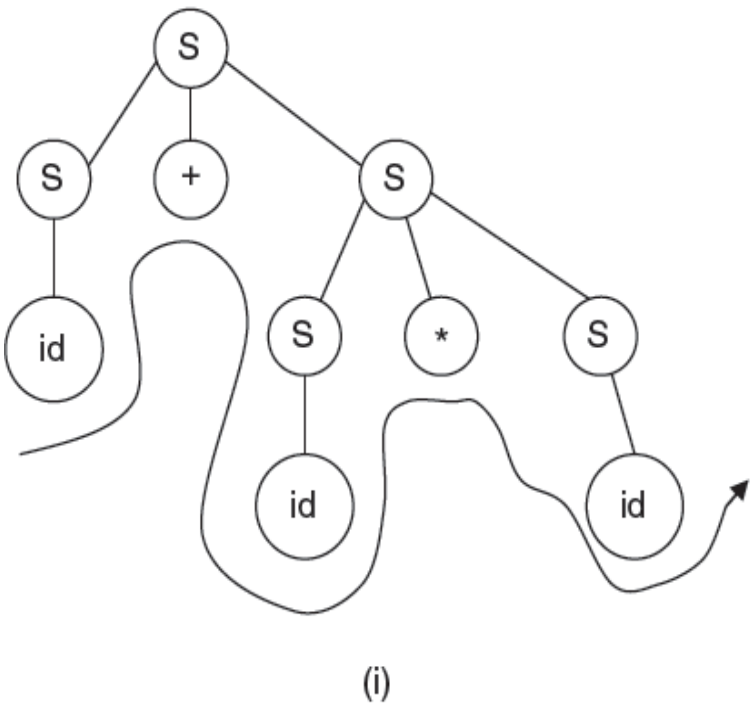
Solution: Let us take a string $id + id * id$.

The string can be generated in the following ways.

i) $\mathrm{S} \to \mathrm{S} + \underline{\mathrm{S}} \to \underline{\mathrm{S}} + \mathrm{S}*\mathrm{S} \to \mathrm{id} + \underline{\mathrm{S}}*\mathrm{S} \to \mathrm{id} + \mathrm{id}*\underline{\mathrm{S}} \to \mathrm{id} + \mathrm{id}*\mathrm{id}$

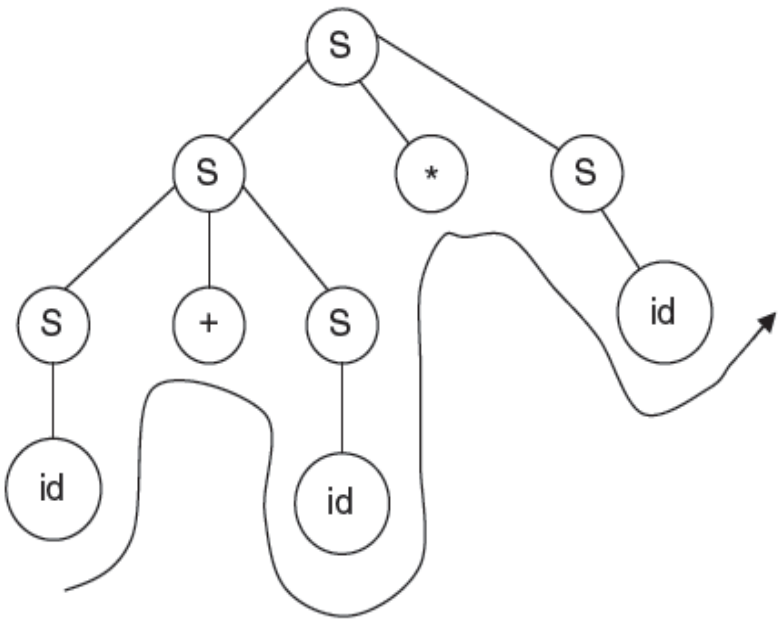
ii) $\mathrm{S} \to \mathrm{S}*\mathrm{S} \to \mathrm{S} + \mathrm{S}*\mathrm{S} \to \mathrm{id} + \mathrm{S}*\mathrm{S} \to \mathrm{id} + \mathrm{id}*\mathrm{S} \to \mathrm{id} + \mathrm{id}*\mathrm{id}$

The parse trees for derivation (i) and (ii) are shown in Figs. 6.7 and 6.8.



(i)

Fig. 6.7 *Parse Tree for Derivation(i)*



(ii)

Fig. 6.8 *Parse Tree for Derivation(ii)*

As we are getting two parse trees for generating a string from the given grammar, the grammar is ambiguous.

Example 6.15 `Prove that the following grammar is ambiguous.`

$$S \rightarrow a/abSb/aAb$$
$$A \rightarrow bS/aAAb$$

Solution: Take a string abababb. The string can be generated in the following ways.

i) $S \rightarrow ab\underline{S}b \rightarrow aba\underline{A}bb \rightarrow abab\underline{S}bb \rightarrow abababb$

ii) $S \rightarrow a\underline{A}b \rightarrow ab\underline{S}b \rightarrow aba\underline{A}bb \rightarrow abab\underline{S}bb \rightarrow abababb$

The parse trees for (i) and (ii) are given in Fig. 6.9.

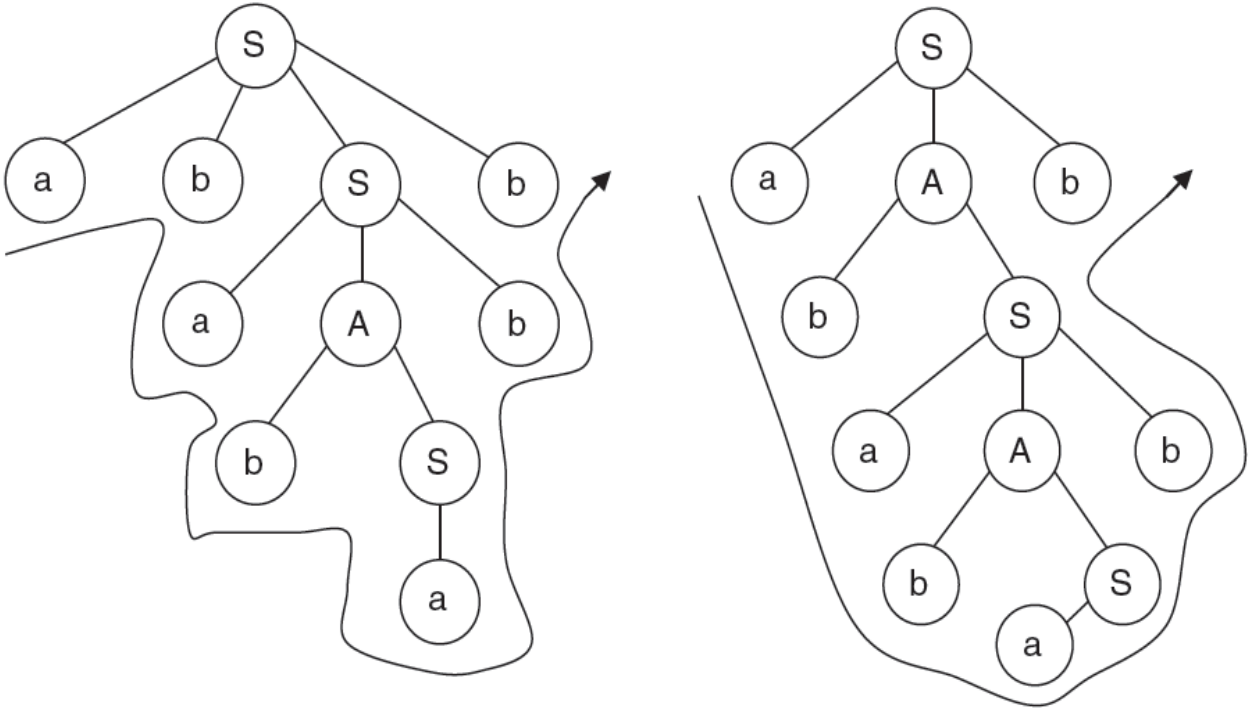


Fig. 6.9 *Parse Tree for Derivation (i) and (ii).*

As we are getting two parse trees for generating a string from the given grammar, the grammar is ambiguous.

Example 6.16 Prove that the following grammar is ambiguous.

$$S \to 0Y/01$$
$$X \to 0XY/0$$
$$Y \to XY1/1$$

Solution: Take a string 000111. The string can be derived in the following ways.

i) $S \to 0\underline{Y} \to 0\underline{X}Y1 \to 00\underline{X}YY1 \to 000\underline{Y}Y1 \to 0001\underline{Y}1 \to 000111$

i) $S \to 0\underline{Y} \to 0X\underline{Y}1 \to 0\underline{X}XY11 \to 00\underline{X}Y11 \to 000\underline{Y}11 \to 000111$

The parse trees for (i) and (ii) are shown is Fig. 6.10.

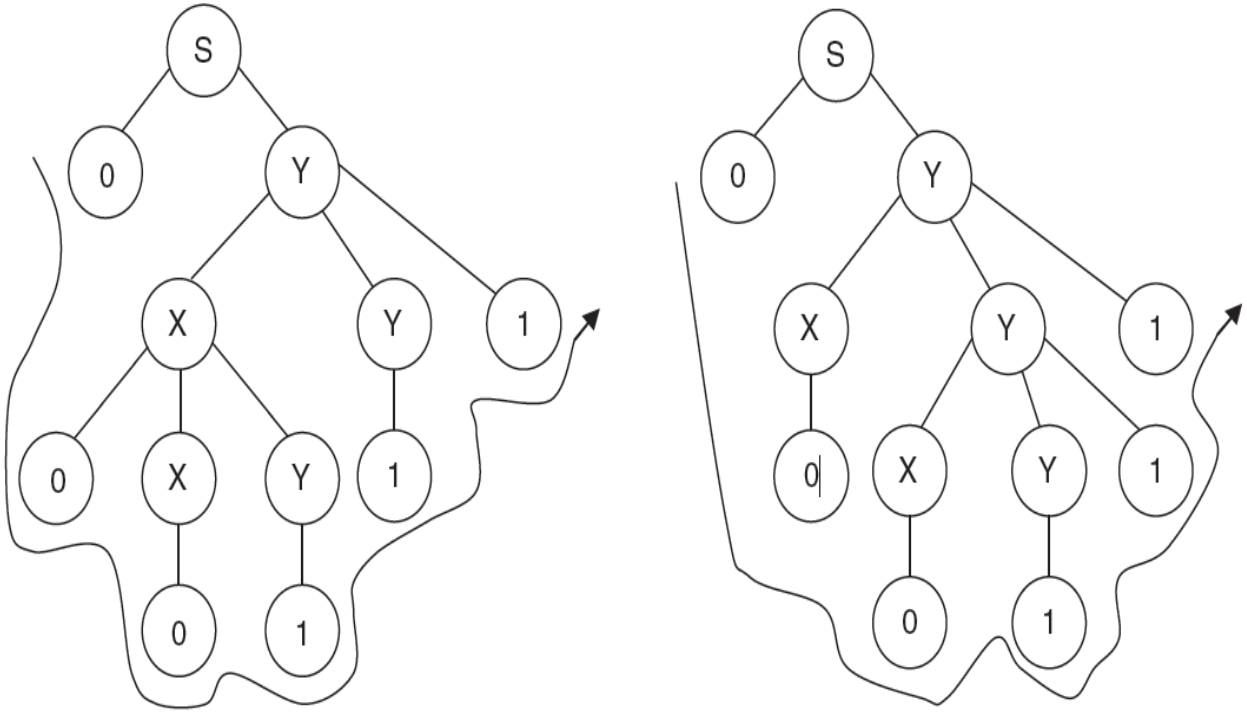


Fig. 6.10 *Parse Tree for Derivation (i) and (ii)*

As we are getting two parse trees for generating a string from the given grammar, the grammar is ambiguous.

In relation to the ambiguity in CFG, there are few more defi nitions. These are related to a CFL. These are:

1. **Ambiguous CFL:** A CFG G is said to be ambiguous if there exists some w $\in$ L(G) that has at least two distinct parse trees.
2. **Inherently Ambiguous CFL:** A CFL L is said to be inherently ambiguous if all its grammars are ambiguous.
3. **Unambiguous CFL:** If L is a CFL for which there exists an unambiguous grammar, then L is said to be unambiguous.

(Even if one grammar for L is unambiguous, then L is an unambiguous language.)

Ambiguous grammar creates problem. Let us take an example.

For a grammar G, the production rule is

$$E \rightarrow E + E/E * E/a.$$

From here, we have to construct $a + a * a$.

The string can be generated in two different ways

i) $E \to E + E \to E + E * E \to a + E * E \to a + a * E \to a + a * a$

ii) $E \to E * E \to E + E * E \to a + E * E \to a + a * E \to a + a * a$

So, for these two cases two parse trees are generated as shown in Fig. 6.11.

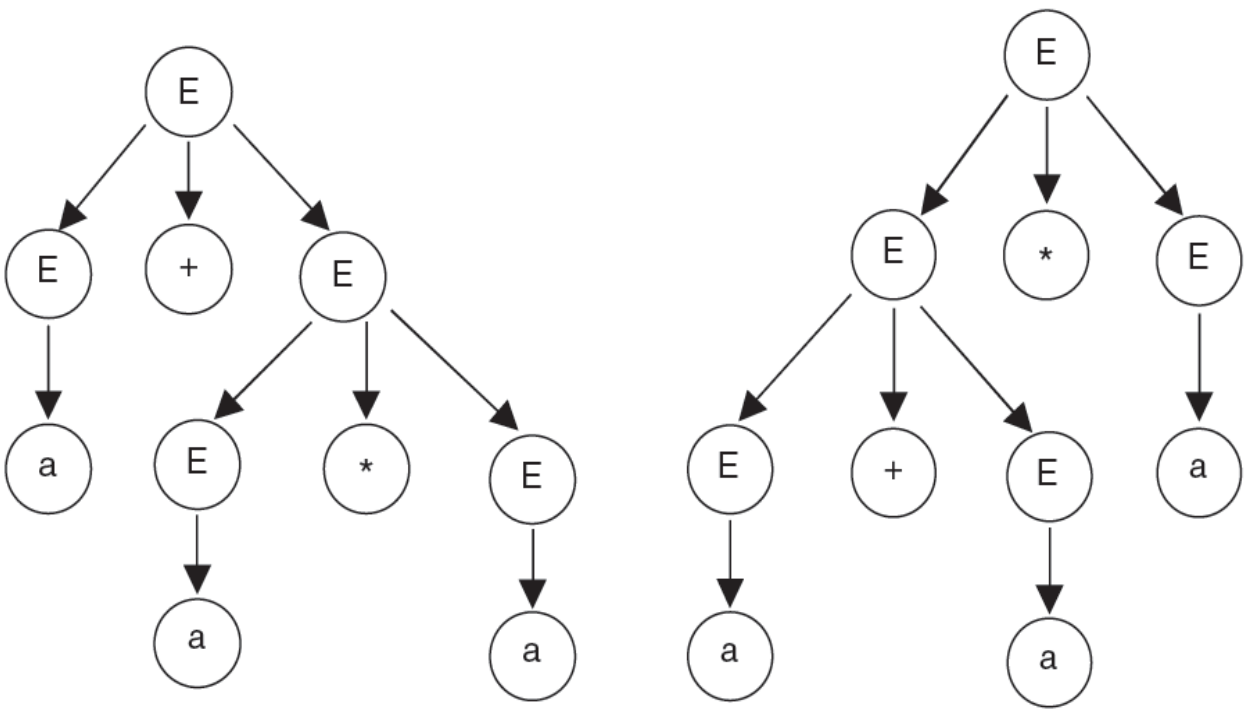


Fig. 6.11 *Parse Tree for Derivation (i) and (ii)*

So, the grammar is ambiguous.

In the place of 'a', put '2'. So, the derivations will be

i) $E \rightarrow E + E \rightarrow E + E * E \rightarrow 2 + E * E \rightarrow 2 + 2 * E \rightarrow 2 + 2 * 2$

ii) $E \rightarrow E * E \rightarrow E + E * E \rightarrow 2 + E * E \rightarrow 2 + 2 * E \rightarrow 2 + 2 * 2$

Up to this step, both of them seem the same. But the real problem is in the parse tree as shown in Fig. 6.12.

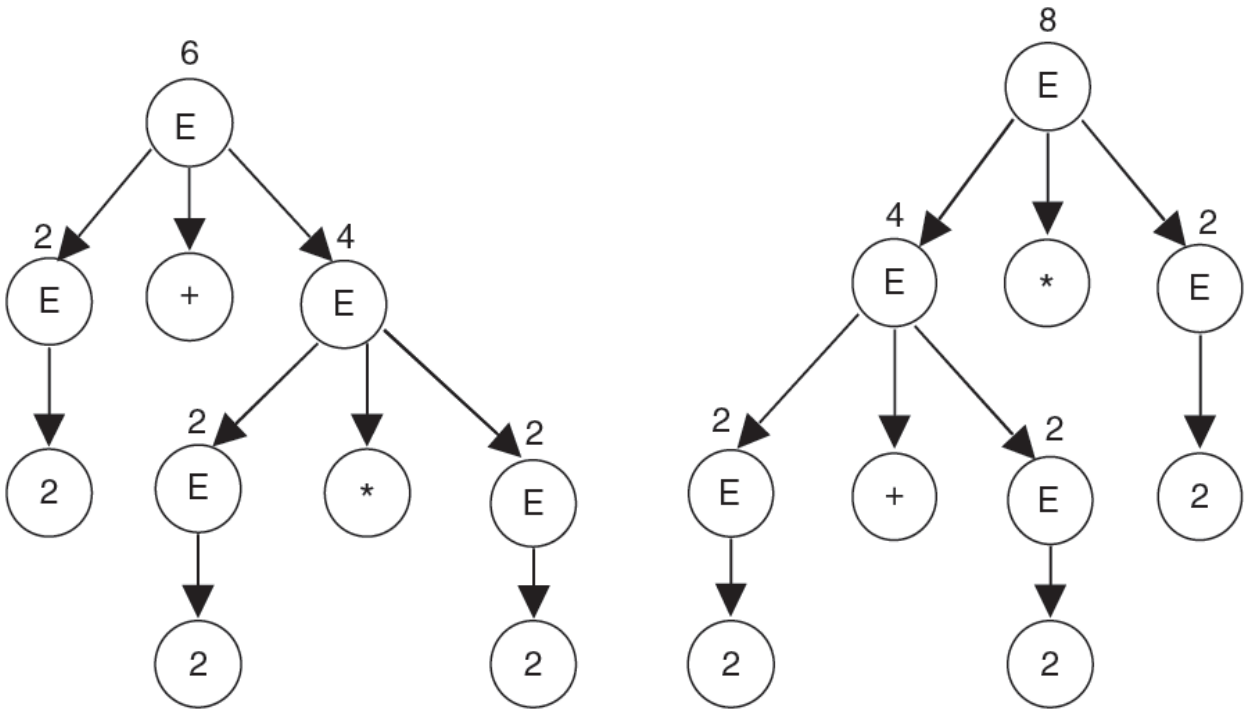


Fig. 6.12 *Annotated Parse Tree for (i) and (ii)*

The correct result is $2 + 2 * 2 = 2 + 4 = 6$ (according to the rules of mathematics $*$ has higher precedence over $+$).