CSE 322

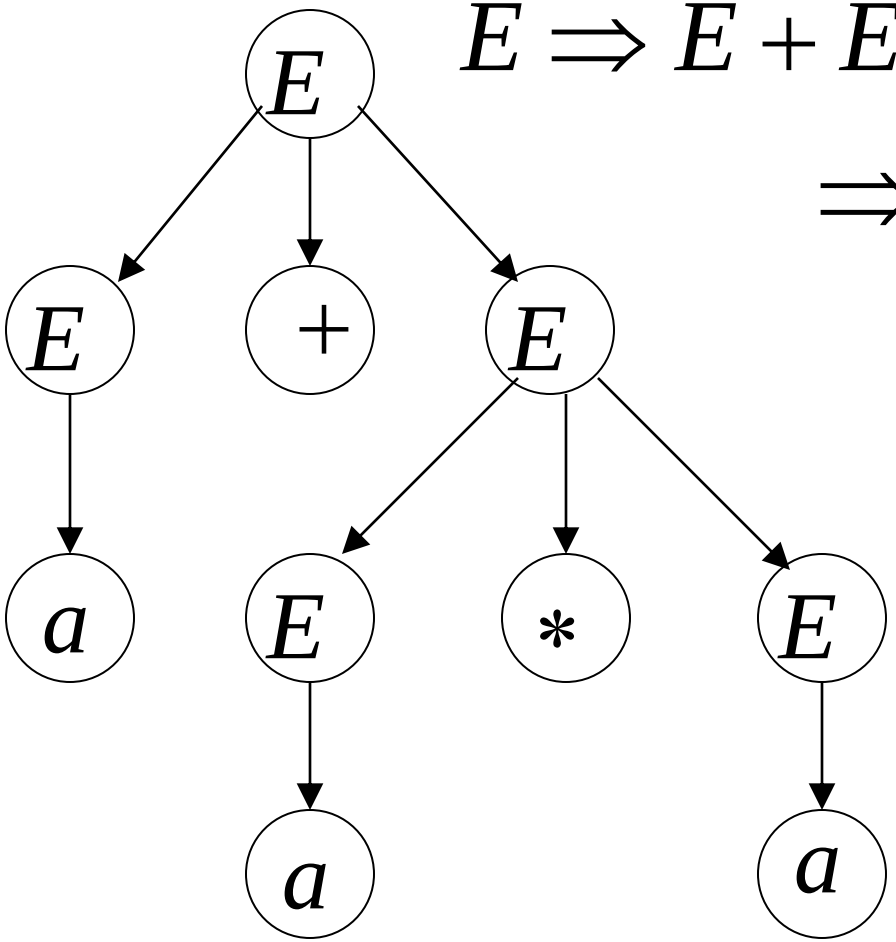
AMBIGUITY IN GRAMMAR

Lecture #26

Ambiguity

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

$$a + a * a$$



$$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E * E$$

$$\Rightarrow a + a * E \Rightarrow a + a * a$$

leftmost derivation

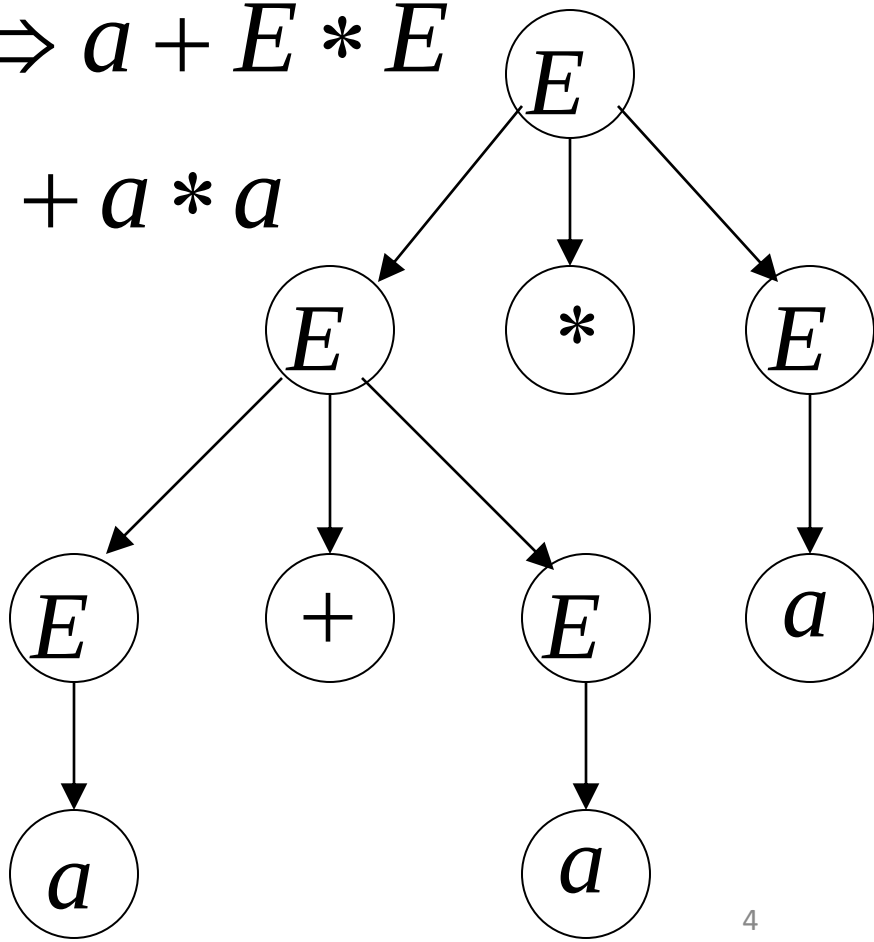
$$E \rightarrow E + E \mid E * E \mid (E) \mid a$$

$$a + a * a$$

$$E \Rightarrow E * E \Rightarrow E + E * E \Rightarrow a + E * E$$

$$\Rightarrow a + a * E \Rightarrow a + a * a$$

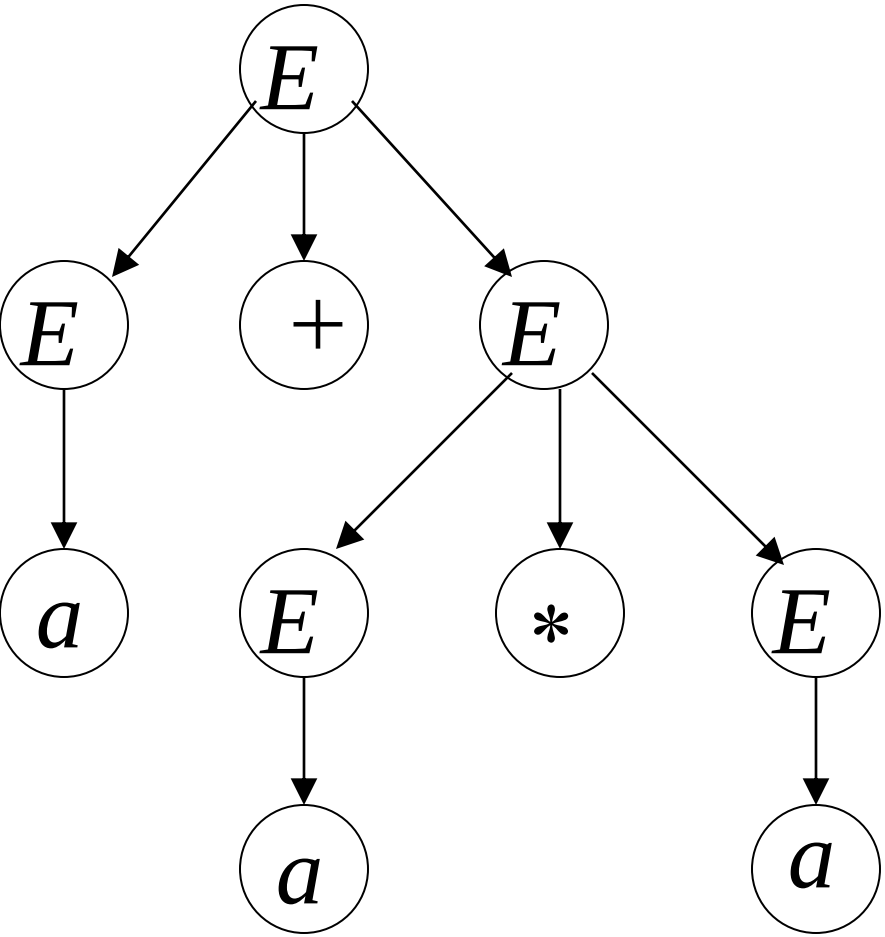
leftmost derivation

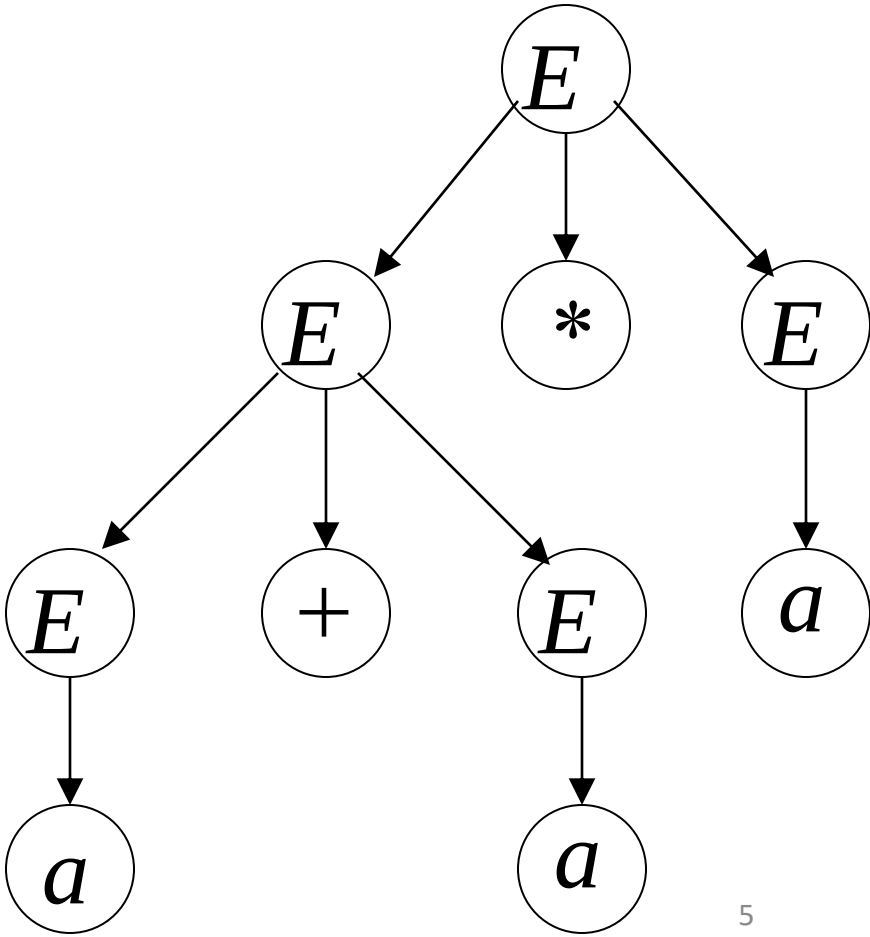


$$E \rightarrow E + E \mid E * E \mid (E) \mid a$$

$$a + a * a$$

Two derivation trees

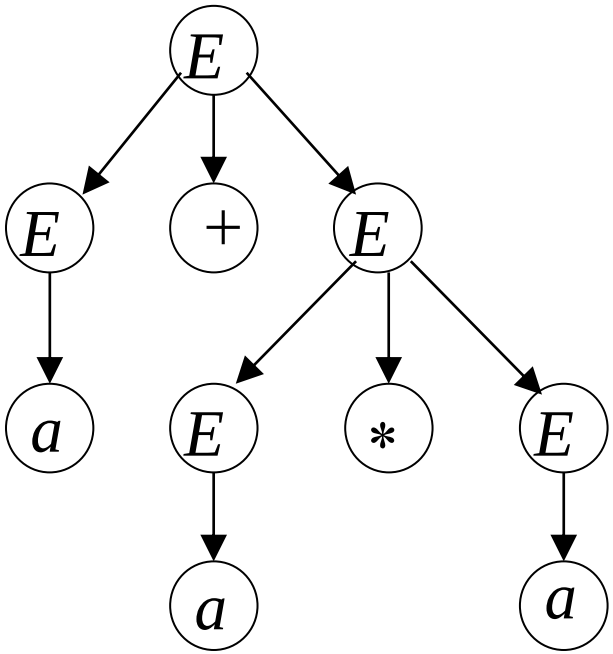


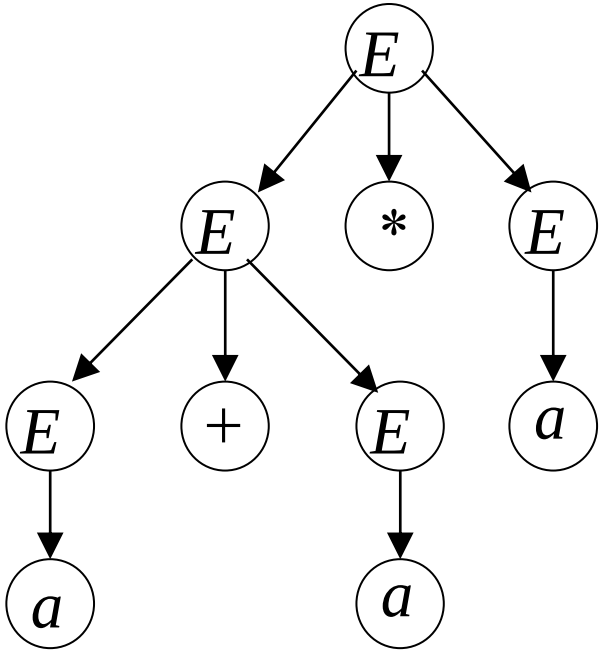


The grammar is ambiguous

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

string $a + a * a$ has two derivation trees





The grammar is ambiguous

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

string $a + a * a$ has two leftmost derivations

$$E \Rightarrow E + E \Rightarrow a + E \Rightarrow a + E * E$$
$$\Rightarrow a + a * E \Rightarrow a + a * a$$

$$E \Rightarrow E * E \Rightarrow E + E * E \Rightarrow a + E * E$$
$$\Rightarrow a + a * E \Rightarrow a + a * a$$

Definition:

A context-free grammar is **ambiguous**

if some string has:

two or more derivation trees

$$w \in L(G)$$

In other words:

A context-free grammar G is **ambiguous**

if some string has:

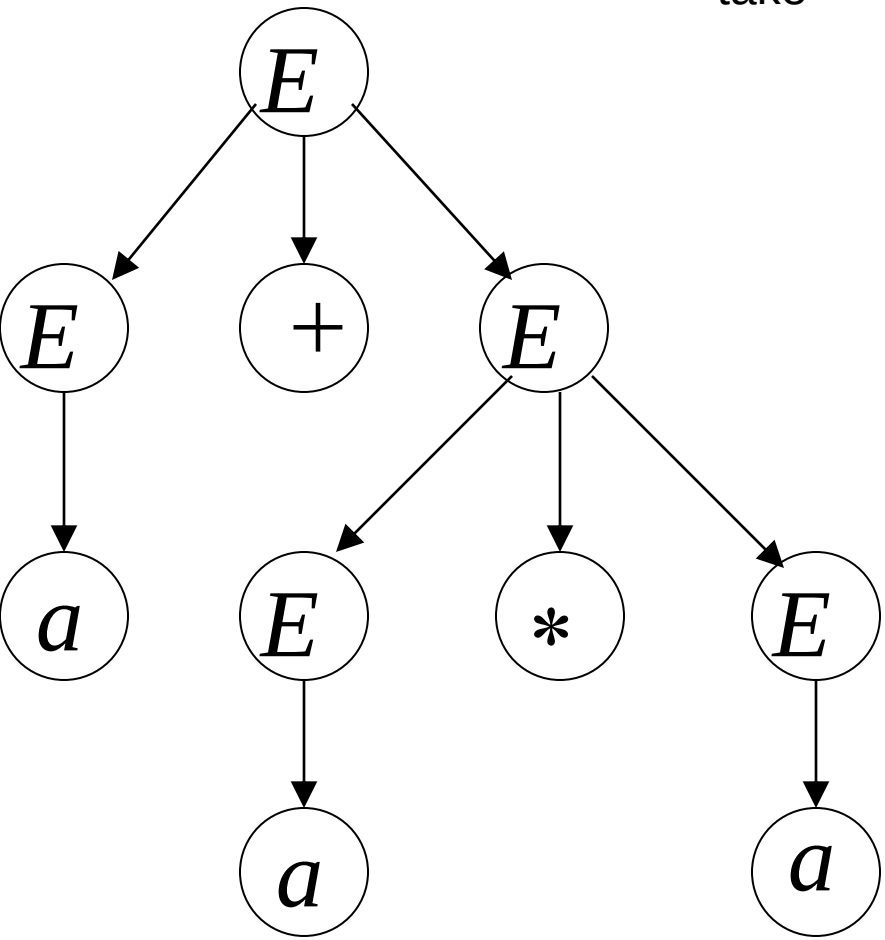
two or more leftmost derivations

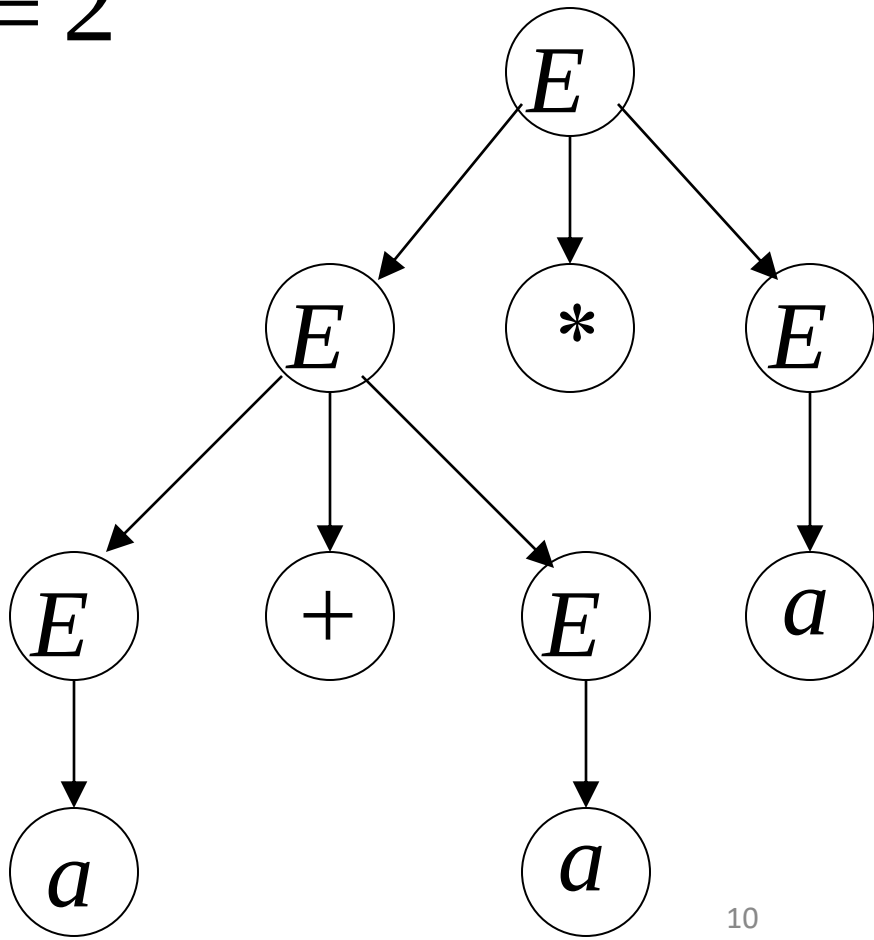
$$w \in L(G)$$

(or rightmost)

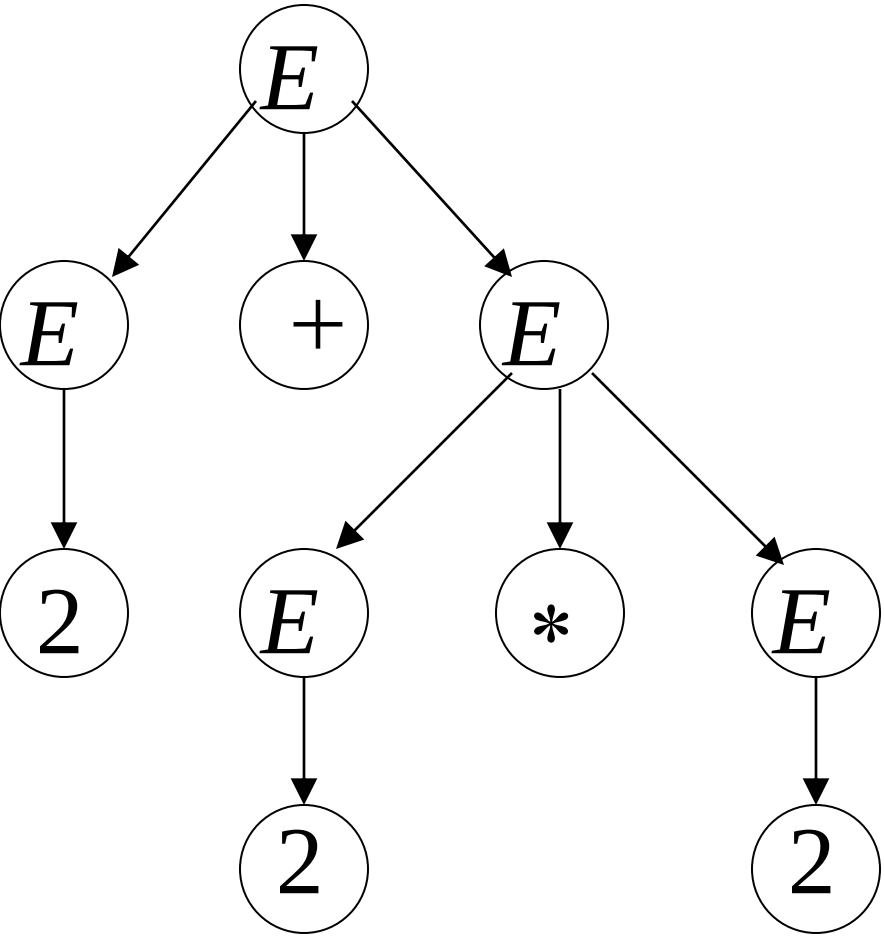
$$a + a * a$$

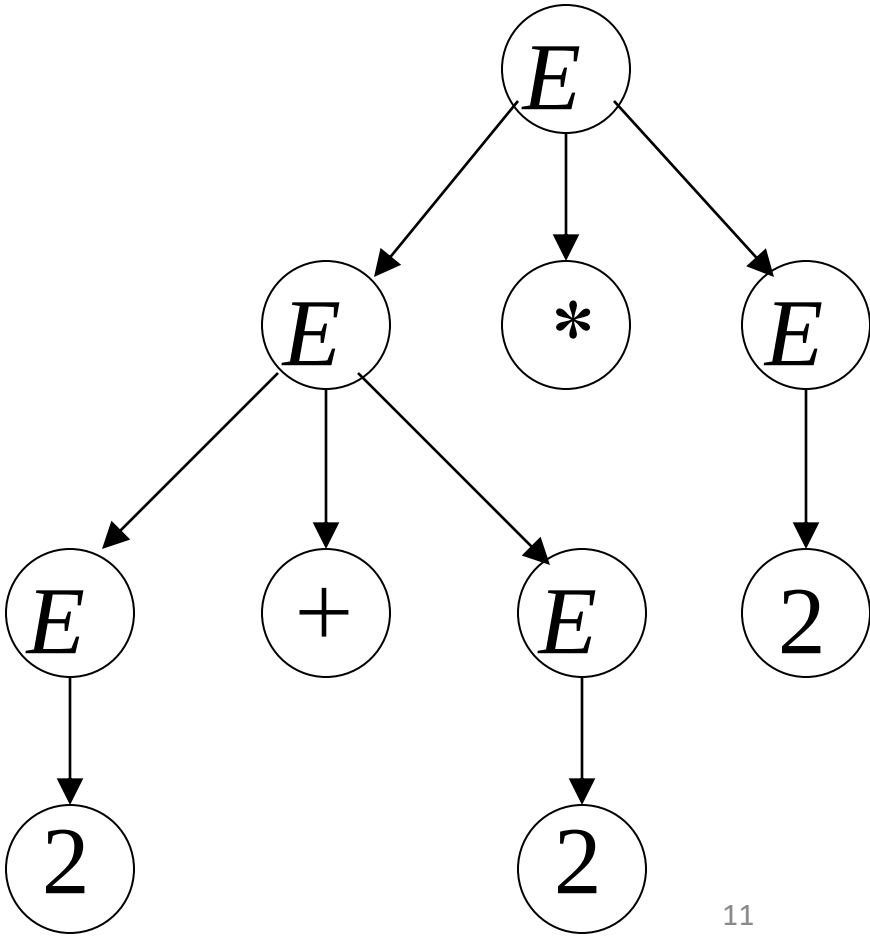
take $a = 2$



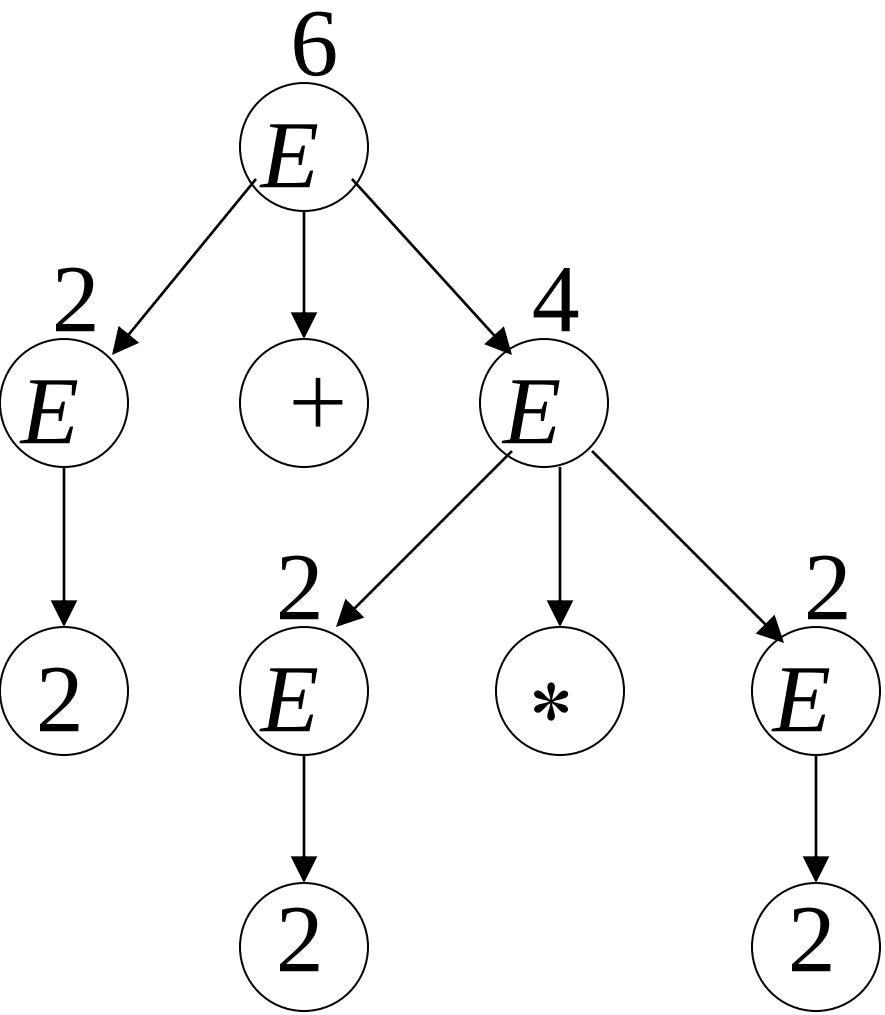


$$2 + 2 * 2$$

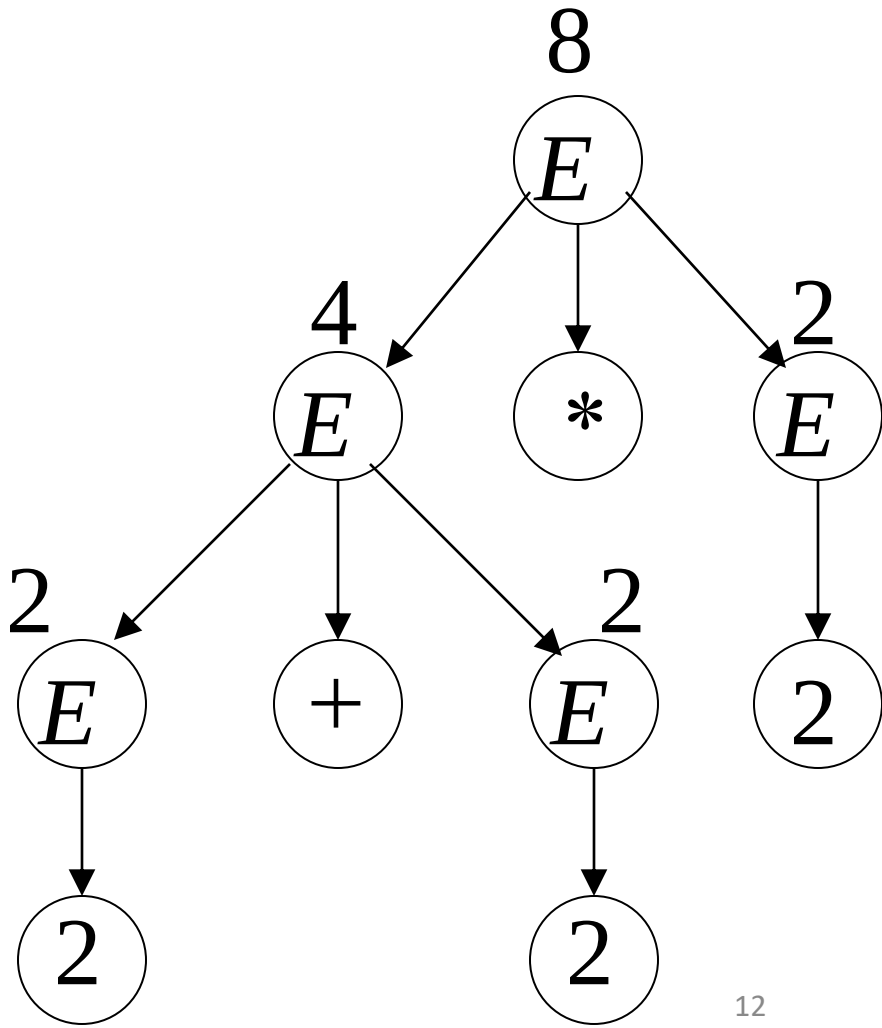




$2+2*2=6$

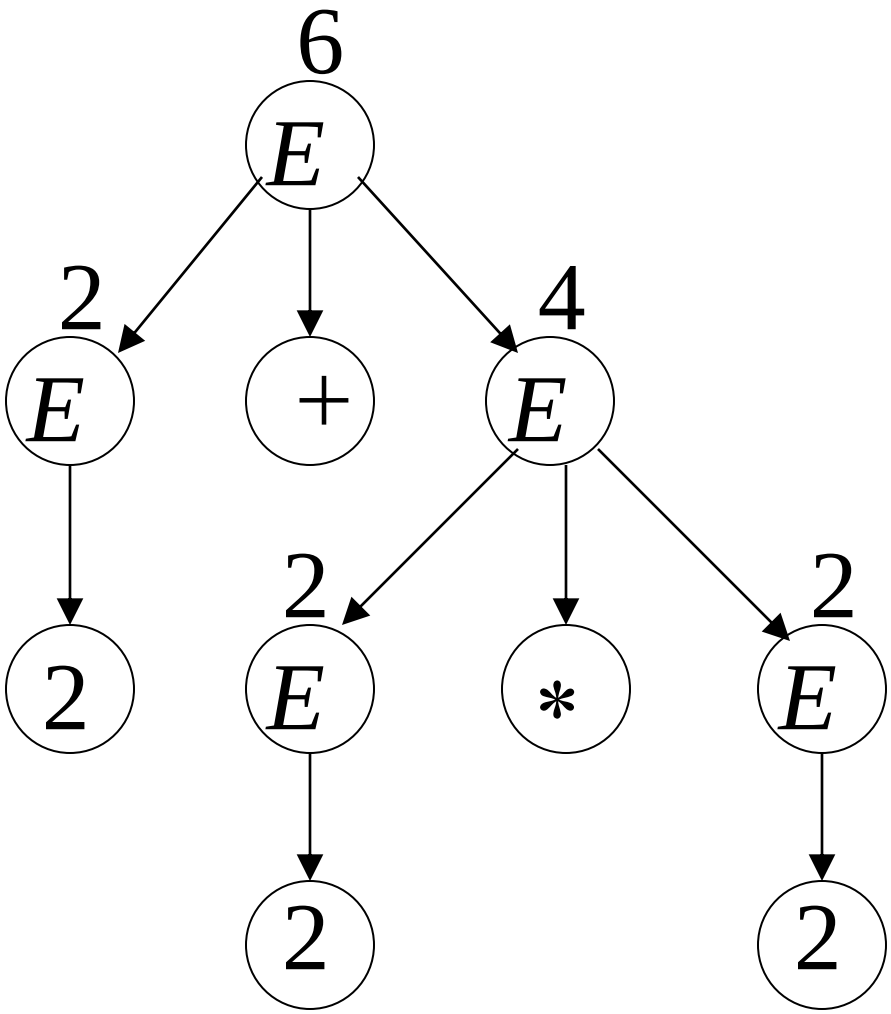


$2+2*2=8$



Correct result:

$$2 + 2 * 2 = 6$$



- Ambiguity is **bad** for programming languages

- We want to remove ambiguity

We fix the ambiguous grammar:

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

New non-ambiguous grammar:

$$E \rightarrow E + T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow F$$

$$F \rightarrow (E)$$

$$F \rightarrow a$$

$$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T \Rightarrow a+T \Rightarrow a+T*F$$

$$\Rightarrow a+F*F \Rightarrow a+a*F \Rightarrow a+a*a$$

$$E \rightarrow E+T$$

$$E \rightarrow T$$

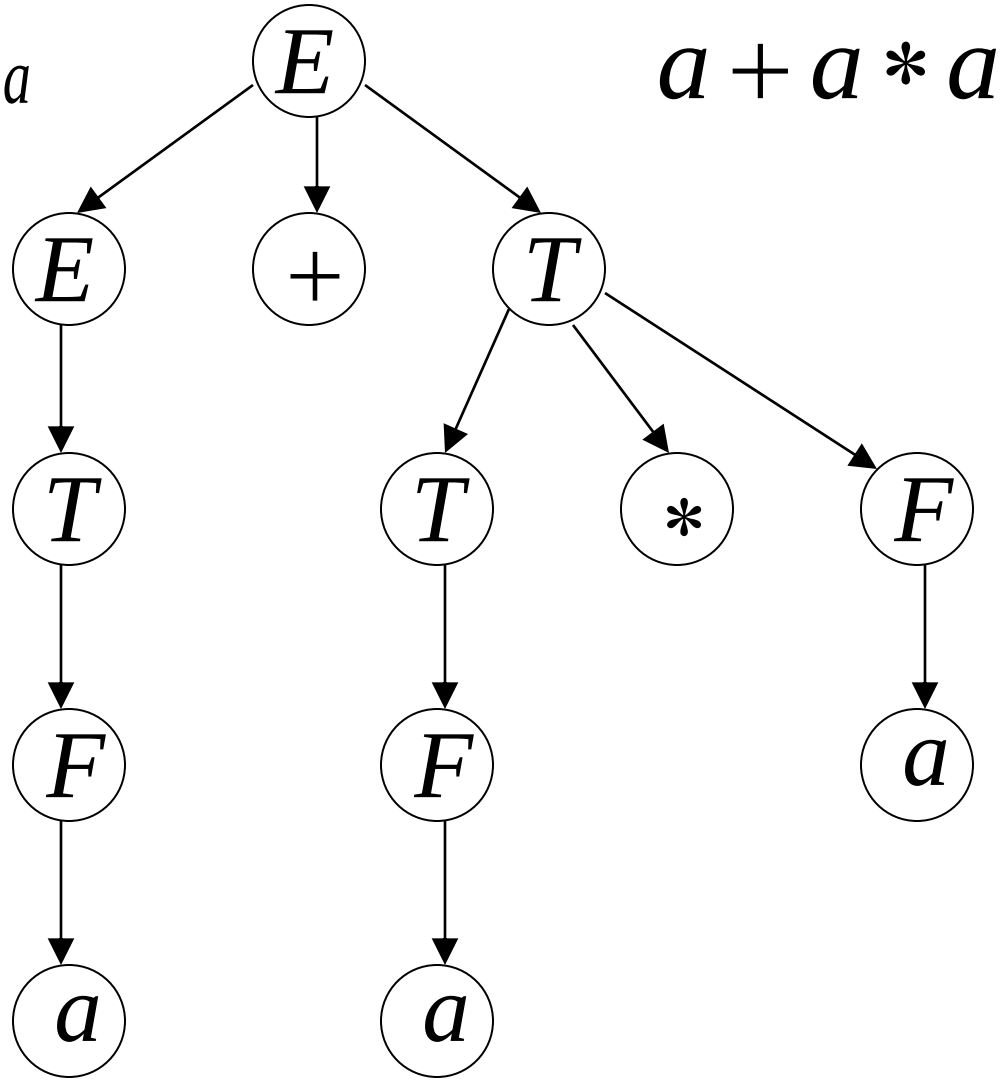
$$T \rightarrow T*F$$

$$T \rightarrow F$$

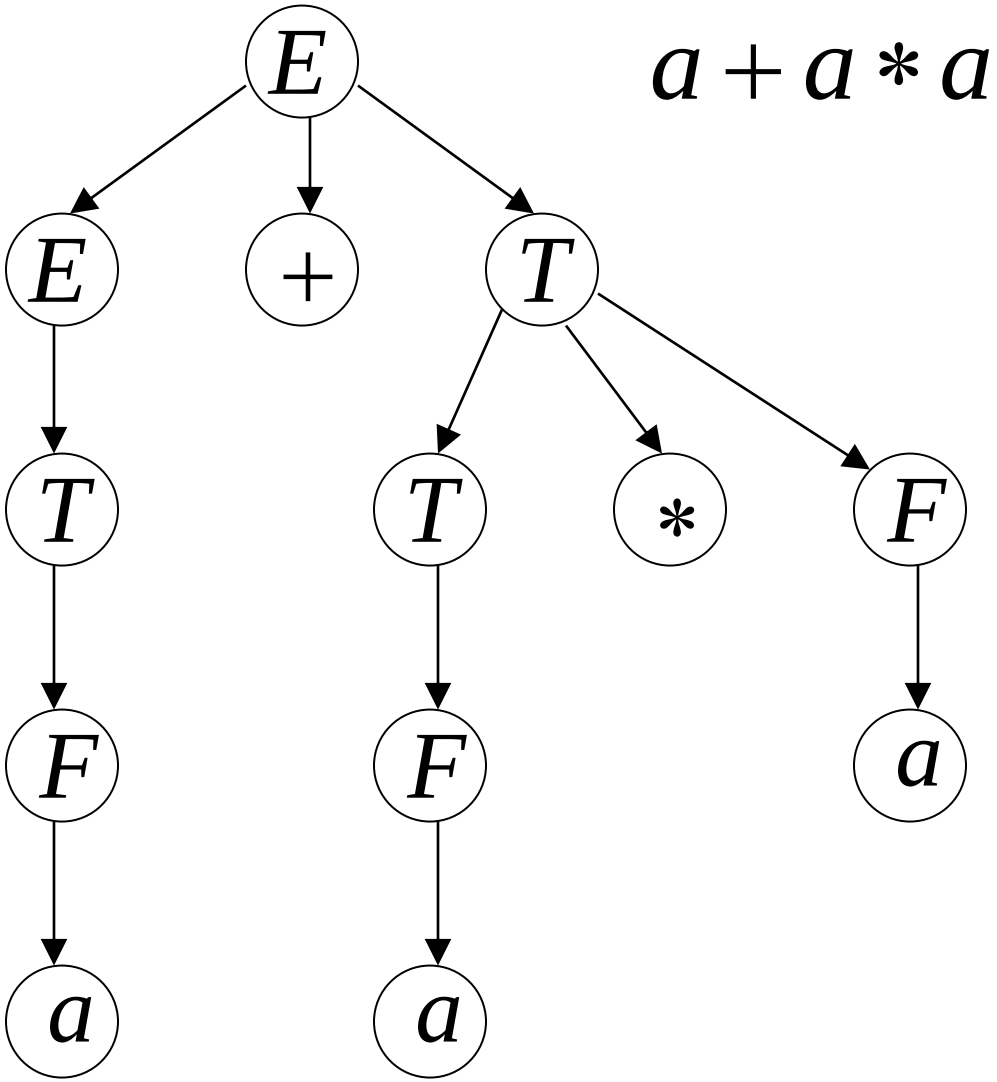
$$F \rightarrow (E)$$

$$F \rightarrow a$$

$$a+a*a$$



Unique derivation tree



The grammar : G

$$E \rightarrow E + T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow F$$

$$F \rightarrow (E)$$

$$F \rightarrow a$$

is non-ambiguous:

$$w \in L(G)$$

Every string has
a unique derivation tree

Inherent Ambiguity

- Some context free languages
- have only ambiguous grammars

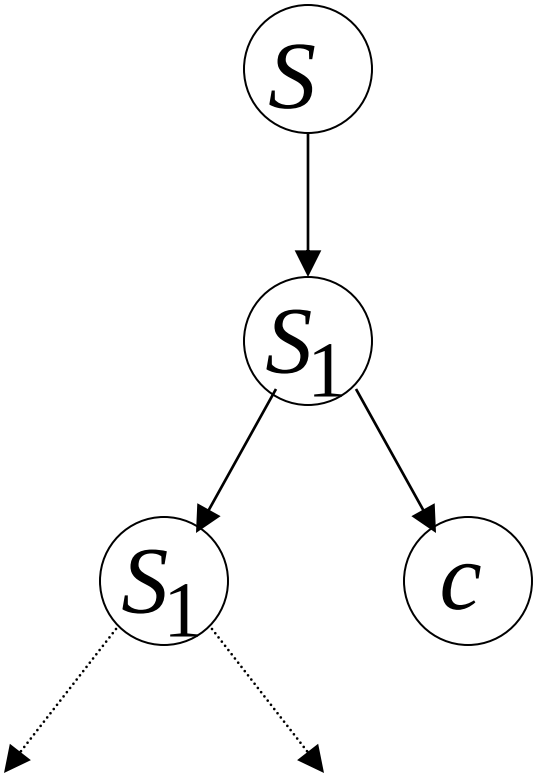
Example:

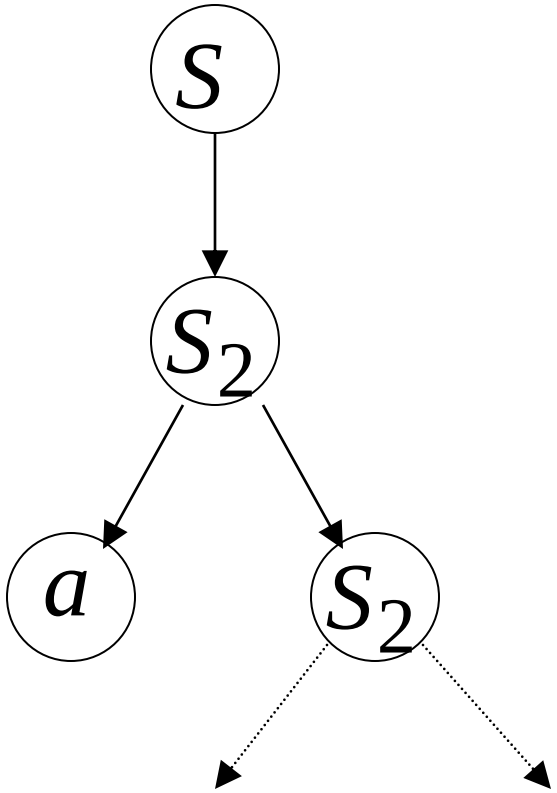
$$L = \{a^n b^n c^m\} \cup \{a^n b^m c^m\}$$

$$S \to S_1 \mid S_2 \qquad S_1 \to S_1 c \mid A \qquad S_2 \to aS_2 \mid B$$

$$A \to aAb \mid \lambda \qquad B \to bBc \mid \lambda$$

The string has two derivation trees $a^n b^n c^n$





Ambiguity in context free grammar

A terminal string $w \in L(G)$ is ambiguous if there exist two or more derivation trees for w (or there exist two or more leftmost derivations of w).

PROBLEM

If G is the grammar $S \rightarrow SbS \mid a$, show that G is ambiguous.

SOLUTION