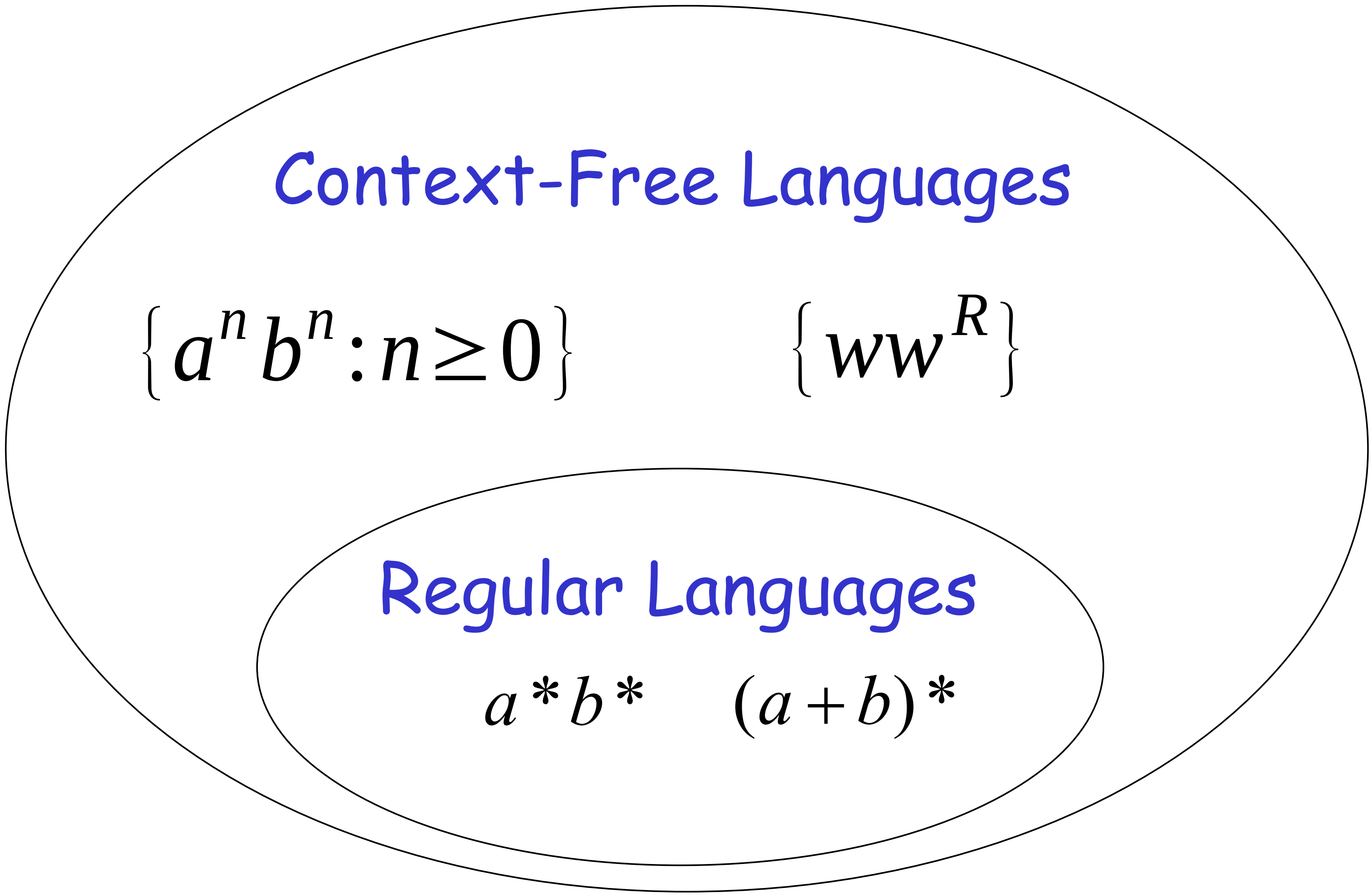
CS116-Automata Theory and Formal Languages

Lecture 7

Context-Free Languages

Computer Science Department

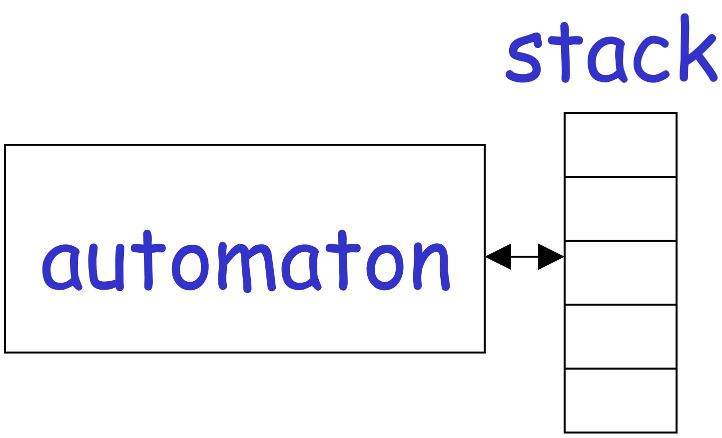
1st Semester 2025-2026


Context-Free Languages
$\{a^n b^n : n \geq 0\}$
$\{ww^R\}$
Regular Languages
a^*b^*
$(a+b)^*$

Context-Free Languages

Context-Free Grammars

Pushdown Automata



Context-Free Grammars

Grammars

Grammars express languages

Example: the English language grammar

$$\langle sentence \rangle \to \langle noun\_phrase \rangle \ \langle predicate \rangle$$

$$\langle noun\_phrase \rangle \to \langle article \rangle \ \langle noun \rangle$$

$$\langle predicate \rangle \to \langle verb \rangle$$

$$\langle article \rangle \to a$$

$$\langle article \rangle \to the$$

$$\langle noun \rangle \to cat$$

$$\langle noun \rangle \to dog$$

$$\langle verb \rangle \to runs$$

$$\langle verb \rangle \to sleeps$$

Derivation of string "the dog walks":

$$
\begin{aligned}
\langle sentence \rangle &\Rightarrow \langle noun\_phrase \rangle \ \langle predicate \rangle \\
&\Rightarrow \langle noun\_phrase \rangle \ \langle verb \rangle \\
&\Rightarrow \langle article \rangle \ \langle noun \rangle \ \langle verb \rangle \\
&\Rightarrow the \ \langle noun \rangle \ \langle verb \rangle \\
&\Rightarrow the \ \ dog \ \langle verb \rangle \\
&\Rightarrow the \ \ dog \ \ sleeps
\end{aligned}
$$

Derivation of string "a cat runs":

$$\begin{aligned}
\langle sentence \rangle &\Rightarrow \langle noun_phrase \rangle \; \langle predicate \rangle \\
&\Rightarrow \langle noun_phrase \rangle \; \langle verb \rangle \\
&\Rightarrow \langle article \rangle \; \langle noun \rangle \; \langle verb \rangle \\
&\Rightarrow a \;\; \langle noun \rangle \; \langle verb \rangle \\
&\Rightarrow a \;\; cat \;\; \langle verb \rangle \\
&\Rightarrow a \;\; cat \;\; runs
\end{aligned}$$

Language of the grammar:

L = { "a cat runs",
"a cat sleeps",
"the cat runs",
"the cat sleeps",
"a dog runs",
"a dog sleeps",
"the dog runs",
"the dog sleeps" }

Productions

$$\langle noun \rangle \rightarrow \overbrace{cat}^{\text{Sequence of Terminals (symbols)}}$$

$$\langle sentence \rangle \rightarrow \underbrace{\langle noun\ phrase \rangle \langle predicate \rangle}_{\text{Sequence of Variables}}$$

Variables

Another Example

Sequence of
terminals and variables

Grammar:

$$S \to \overbrace{aSb}$$
$$S \to \lambda$$

Variable

The right side
may be λ

Grammar:

$$S \to aSb$$
$$S \to \lambda$$

Derivation of string ab :

$$S \Rightarrow aSb \Rightarrow ab$$

$$S \to aSb \qquad\qquad S \to \lambda$$

Grammar:

$$S \to aSb$$
$$S \to \lambda$$

Derivation of string $aabb$:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

$S \to aSb$ $\qquad\qquad$ $S \to \lambda$

Grammar: $S \rightarrow aSb$
$S \rightarrow \lambda$

Other derivations:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$$
$$\Rightarrow aaaaSbbbb \Rightarrow aaaabbbb$$

Grammar:

$$S \rightarrow aSb$$
$$S \rightarrow \lambda$$

Language of the grammar:

$$L = \{a^n b^n : n \geq 0\}$$

A Convenient Notation

We write: $S \overset{*}{\Rightarrow} aaabbb$

for zero or more derivation steps

Instead of:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

In general we write: $w_1 \overset{*}{\Rightarrow} w_n$

If: $w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \cdots \Rightarrow w_n$

in zero or more derivation steps

Trivially: $w \overset{*}{\Rightarrow} w$

Example Grammar

$$S \rightarrow aSb$$
$$S \rightarrow \lambda$$

Possible Derivations

$$S \overset{*}{\Rightarrow} \lambda$$

$$S \overset{*}{\Rightarrow} ab$$

$$S \overset{*}{\Rightarrow} aaabbb$$

$$S \overset{*}{\Rightarrow} aaSbb \overset{*}{\Rightarrow} aaaaaSbbbbb$$

Another convenient notation:

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow \lambda \end{aligned} \quad \Longrightarrow \quad S \rightarrow aSb \mid \lambda$$

$$\begin{aligned} \langle article \rangle &\rightarrow a \\ \langle article \rangle &\rightarrow the \end{aligned} \quad \Longrightarrow \quad \langle article \rangle \rightarrow a \mid the$$

Formal Definitions

Grammar: $G = (V, T, S, P)$

Set of variables

Set of terminal symbols

Start variable

Set of productions

Context-Free Grammar: $G = (V, T, S, P)$

All productions in P are of the form

$$A \rightarrow s$$

Variable

String of variables and terminals

Example of Context-Free Grammar

$$\boxed{S \rightarrow aSb \mid \lambda}$$

productions

$$P = \{ S \rightarrow aSb,\ S \rightarrow \lambda \}$$

$$G = (V, T, S, P)$$

$$V = \{ S \}$$

variables

$$T = \{ a, b \}$$

terminals

start variable

Language of a Grammar:

For a grammar G with start variable S

$$L(G) = \{w: \quad S \overset{*}{\Rightarrow} w, \quad w \in T^*\}$$

String of terminals or λ

Example:

context-free grammar G : $\boxed{S \rightarrow aSb \mid \lambda}$

$$L(G) = \{a^n b^n : \; n \geq 0\}$$

Since, there is derivation

$S \overset{*}{\Rightarrow} a^n b^n$ for any $n \geq 0$

Context-Free Language:

A language L is context-free

if there is a context-free grammar G

with $L = L(G)$

Example:

$$L = \{a^n b^n : \ n \geq 0\}$$

is a context-free language

since context-free grammar G :

$$\boxed{S \rightarrow aSb \mid \lambda}$$

generates $L(G) = L$

Another Example

Context-free grammar G :

$$S \rightarrow aSa \mid bSb \mid \lambda$$

Example derivations:

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abba$$

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow abaSaba \Rightarrow abaaba$$

$$L(G) = \{ww^R : \quad w \in \{a,b\}^*\}$$

Palindromes of even length

Another Example

Context-free grammar G :

$$S \rightarrow aSb \mid SS \mid \lambda$$

Example derivations:

$$S \Rightarrow SS \Rightarrow aSbS \Rightarrow abS \Rightarrow ab$$

$$S \Rightarrow SS \Rightarrow aSbS \Rightarrow abS \Rightarrow abaSb \Rightarrow abab$$

$$L(G) = \{w : n_a(w) = n_b(w), \text{ and } n_a(v) \geq n_b(v) \text{ in any prefix } v\}$$

Describes matched parentheses:

() ((( ))) (( )) $a = ($, $\mathrm{b} =)$

Derivation Order
and
Derivation Trees

Derivation Order

Consider the following example grammar with 5 productions:

1. $S \rightarrow AB$	2. $A \rightarrow aaA$	4. $B \rightarrow Bb$
	3. $A \rightarrow \lambda$	5. $B \rightarrow \lambda$

1. $S \rightarrow AB$ 2. $A \rightarrow aaA$ 4. $B \rightarrow Bb$

3. $A \rightarrow \lambda$ 5. $B \rightarrow \lambda$

Leftmost derivation order of string aab:

$$S \overset{1}{\Rightarrow} AB \overset{2}{\Rightarrow} aaAB \overset{3}{\Rightarrow} aaB \overset{4}{\Rightarrow} aaBb \overset{5}{\Rightarrow} aab$$

At each step, we substitute the leftmost variable

$$1.\ S \rightarrow AB \qquad 2.\ A \rightarrow aaA \qquad 4.\ B \rightarrow Bb$$
$$3.\ A \rightarrow \lambda \qquad 5.\ B \rightarrow \lambda$$

Rightmost derivation order of string aab:

$$S \overset{1}{\Rightarrow} AB \overset{4}{\Rightarrow} ABb \overset{5}{\Rightarrow} Ab \overset{2}{\Rightarrow} aaAb \overset{3}{\Rightarrow} aab$$

At each step, we substitute the rightmost variable

1. $S \rightarrow AB$ 2. $A \rightarrow aaA$ 4. $B \rightarrow Bb$

3. $A \rightarrow \lambda$ 5. $B \rightarrow \lambda$

Leftmost derivation of aab:

$$S \overset{1}{\Rightarrow} AB \overset{2}{\Rightarrow} aaAB \overset{3}{\Rightarrow} aaB \overset{4}{\Rightarrow} aaBb \overset{5}{\Rightarrow} aab$$

Rightmost derivation of aab :

$$S \overset{1}{\Rightarrow} AB \overset{4}{\Rightarrow} ABb \overset{5}{\Rightarrow} Ab \overset{2}{\Rightarrow} aaAb \overset{3}{\Rightarrow} aab$$

Derivation Trees

Consider the same example grammar:

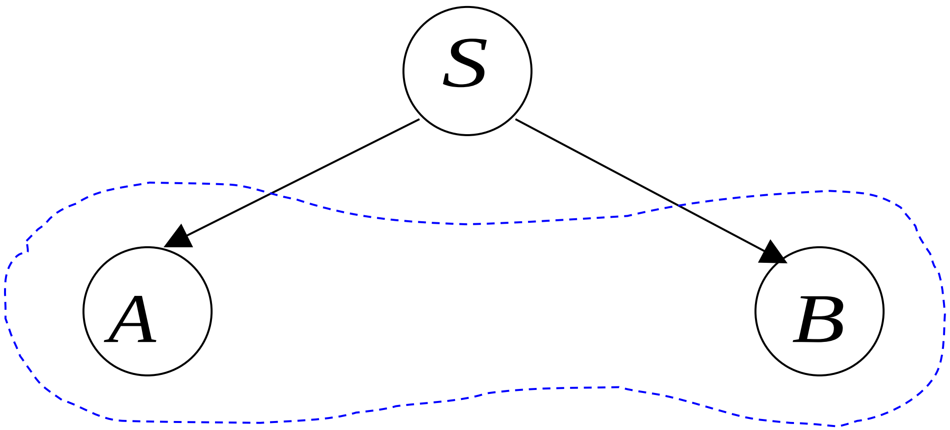
$$S \rightarrow AB \qquad A \rightarrow aaA \mid \lambda \qquad B \rightarrow Bb \mid \lambda$$

And a derivation of aab:

$$S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaABb \Rightarrow aaBb \Rightarrow aab$$

$$S \rightarrow AB \qquad A \rightarrow aaA \mid \lambda \qquad B \rightarrow Bb \mid \lambda$$

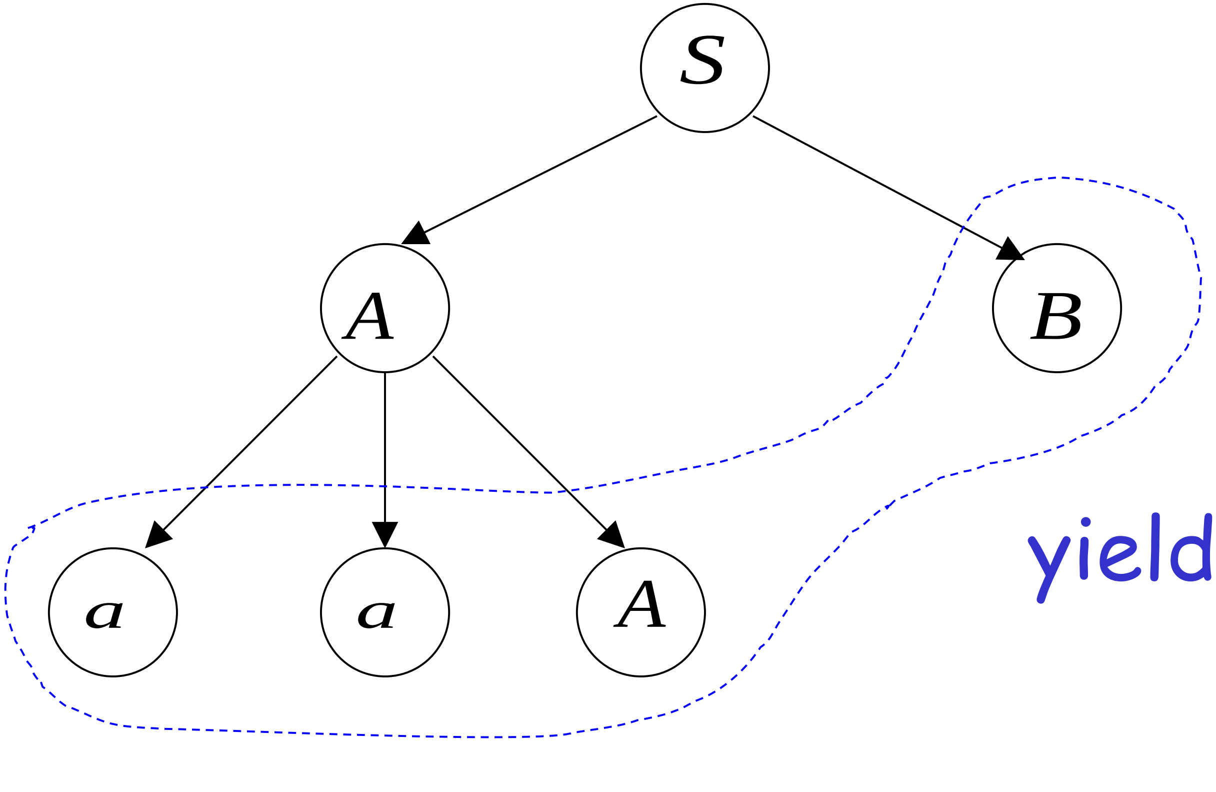
$$S \Rightarrow AB$$



yield AB

$$S \rightarrow AB \qquad A \rightarrow aaA \mid \lambda \qquad B \rightarrow Bb \mid \lambda$$

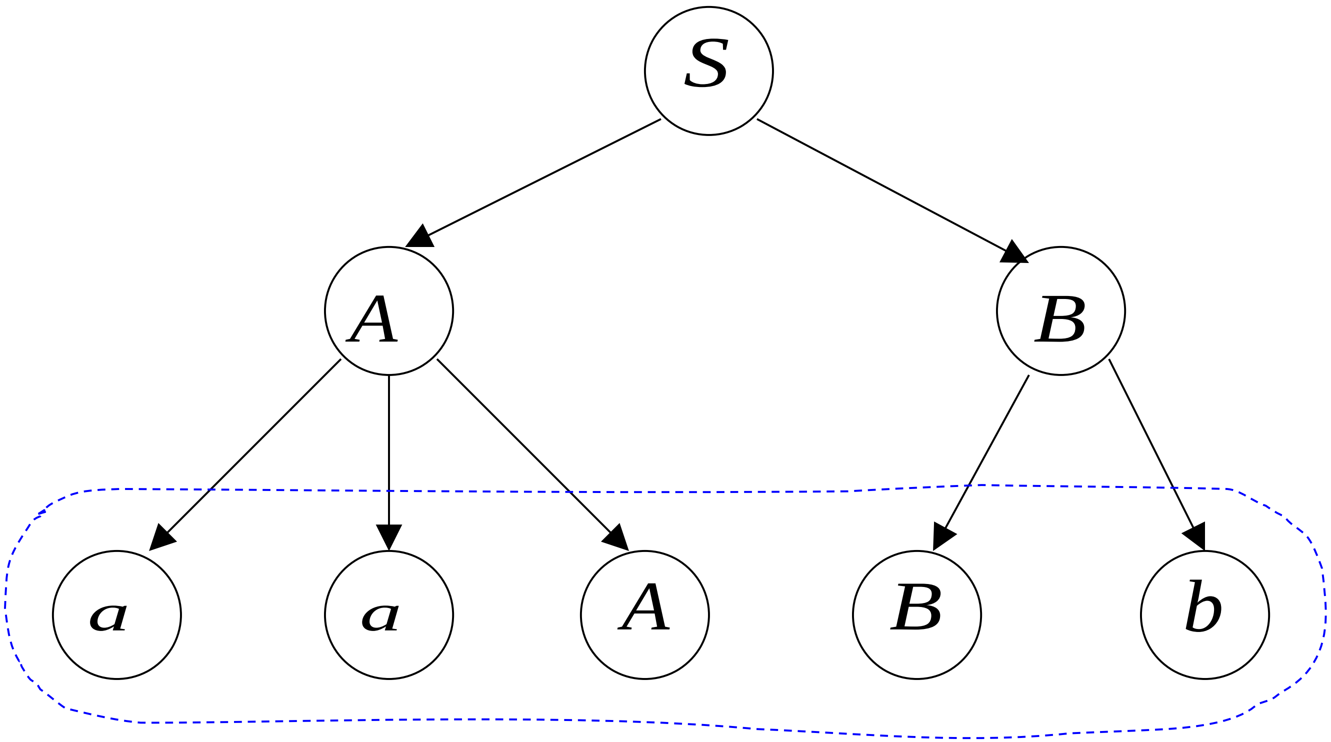
$$S \Rightarrow AB \Rightarrow aaAB$$



yield $aaAB$

$$S \rightarrow AB \qquad A \rightarrow aaA \mid \lambda \qquad B \rightarrow Bb \mid \lambda$$

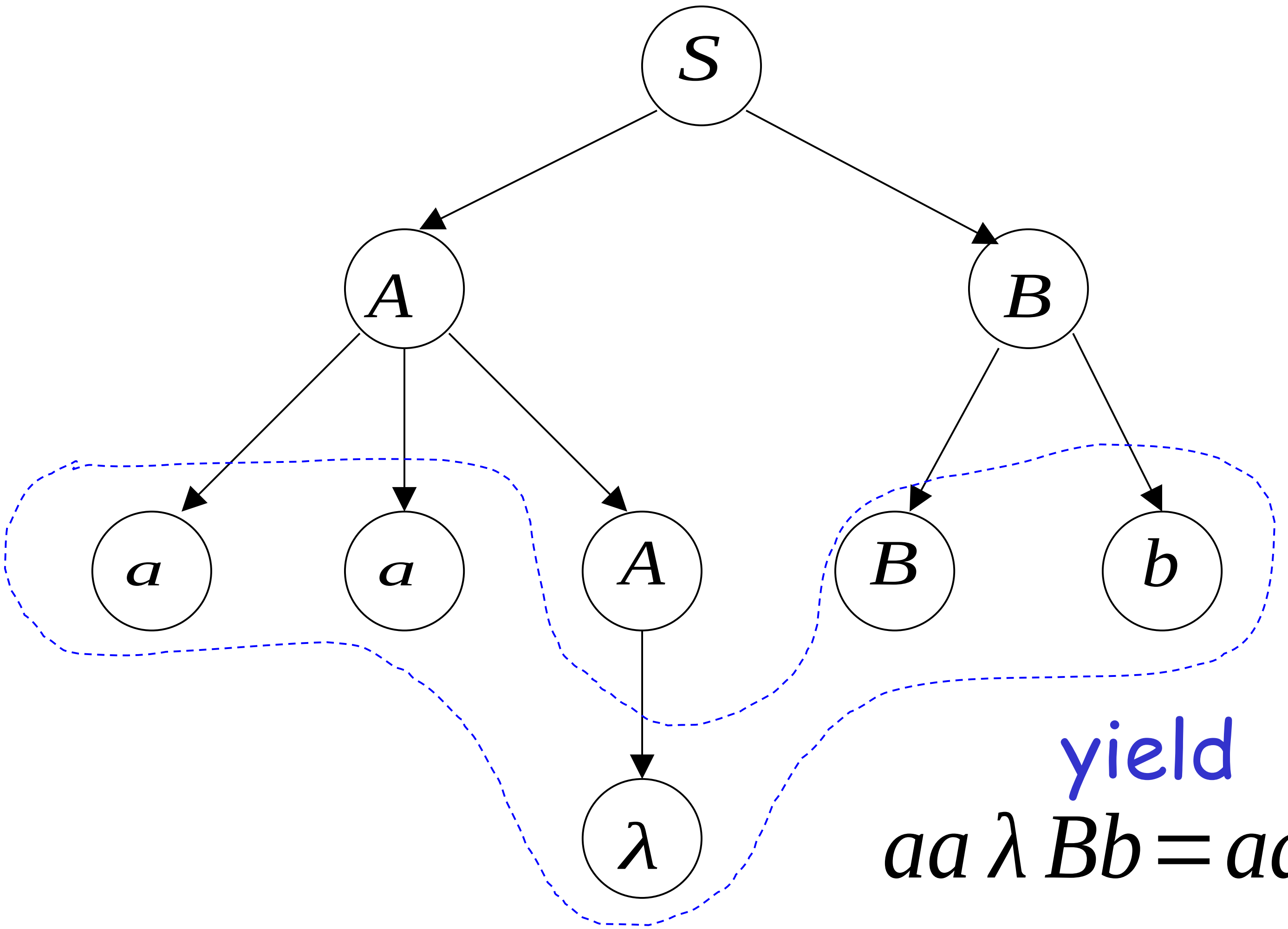
$$S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaABb$$



yield $aaABb$

$$S \rightarrow AB \qquad A \rightarrow aaA \mid \lambda \qquad B \rightarrow Bb \mid \lambda$$

$$S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaABb \Rightarrow aaBb$$



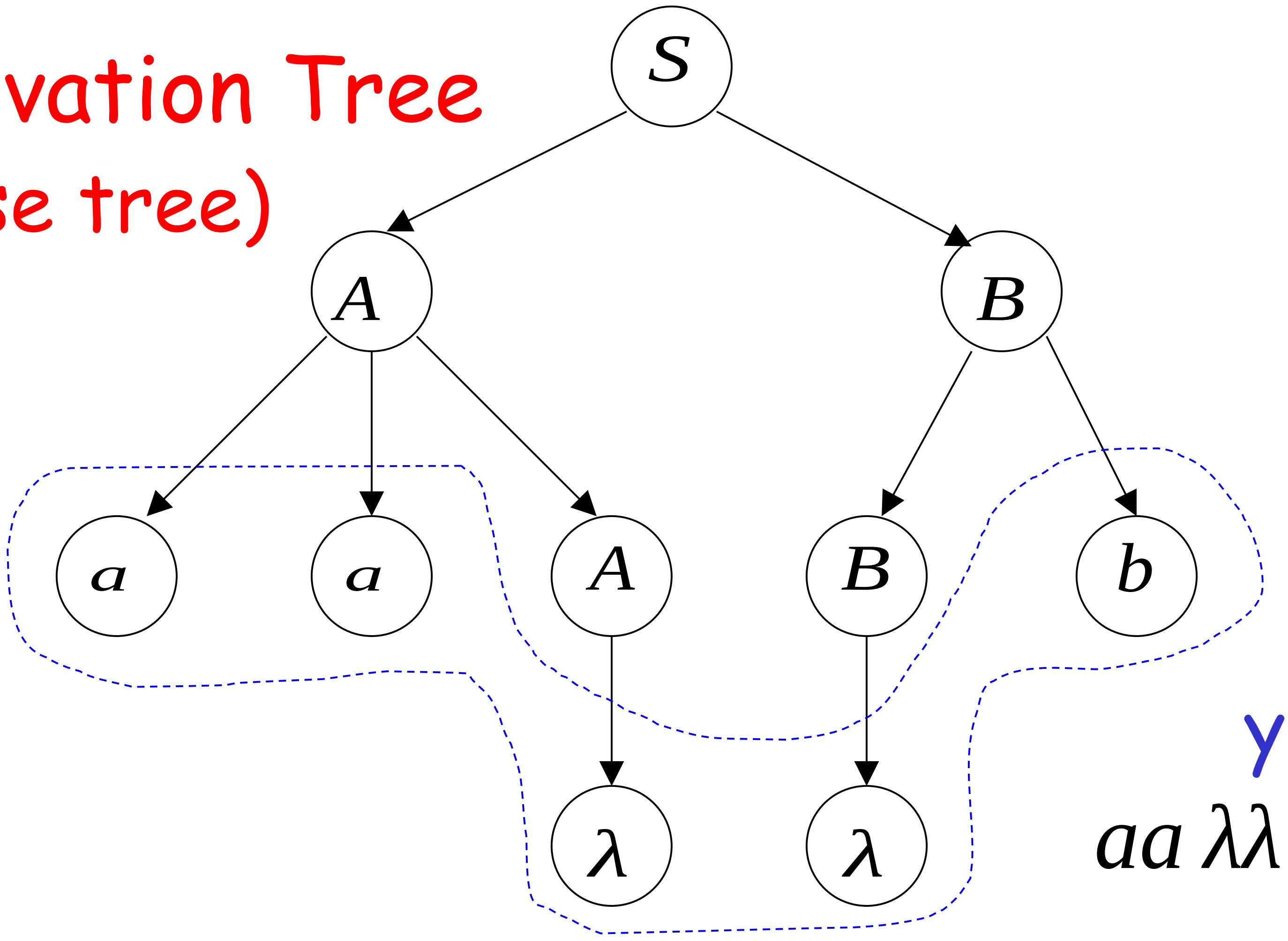
yield

$$aa\lambda Bb = aaBb$$

$$S \to AB \qquad A \to aaA \mid \lambda \qquad B \to Bb \mid \lambda$$

$$S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaABb \Rightarrow aaBb \Rightarrow aab$$

Derivation Tree
(parse tree)



yield

$$aa\lambda\lambda b = aab$$

Sometimes, derivation order doesn't matter

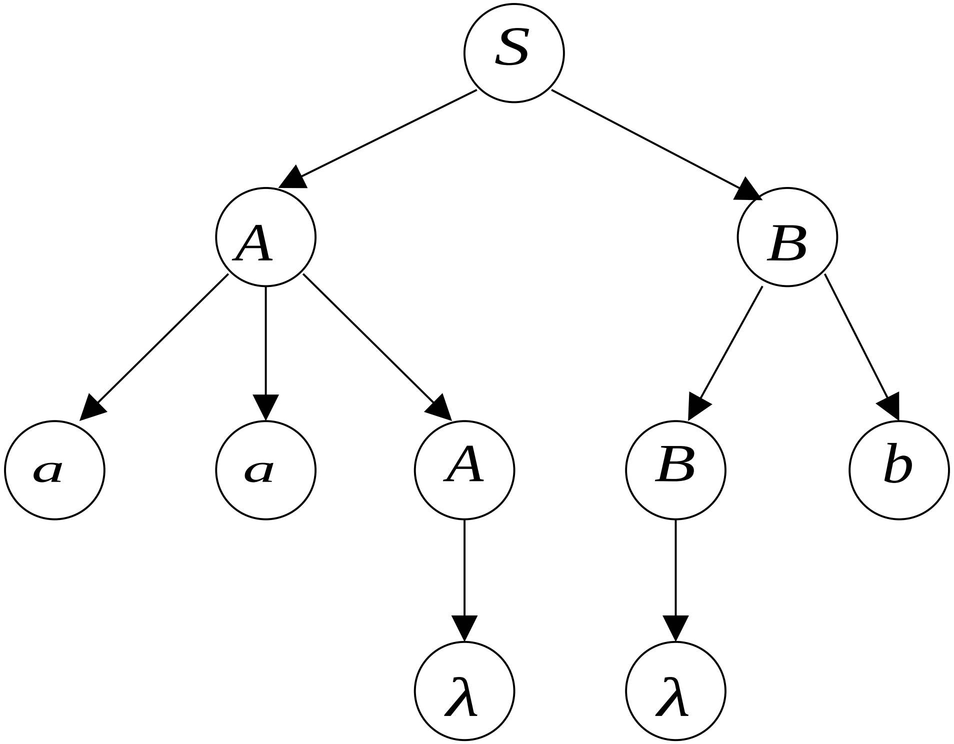
Leftmost derivation:

$$S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaB \Rightarrow aaBb \Rightarrow aab$$

Rightmost derivation:

$$S \Rightarrow AB \Rightarrow ABb \Rightarrow Ab \Rightarrow aaAb \Rightarrow aab$$

Give same derivation tree



Ambiguity

Grammar for mathematical expressions

$$\boxed{E \rightarrow E+E \mid E*E \mid (E) \mid a}$$

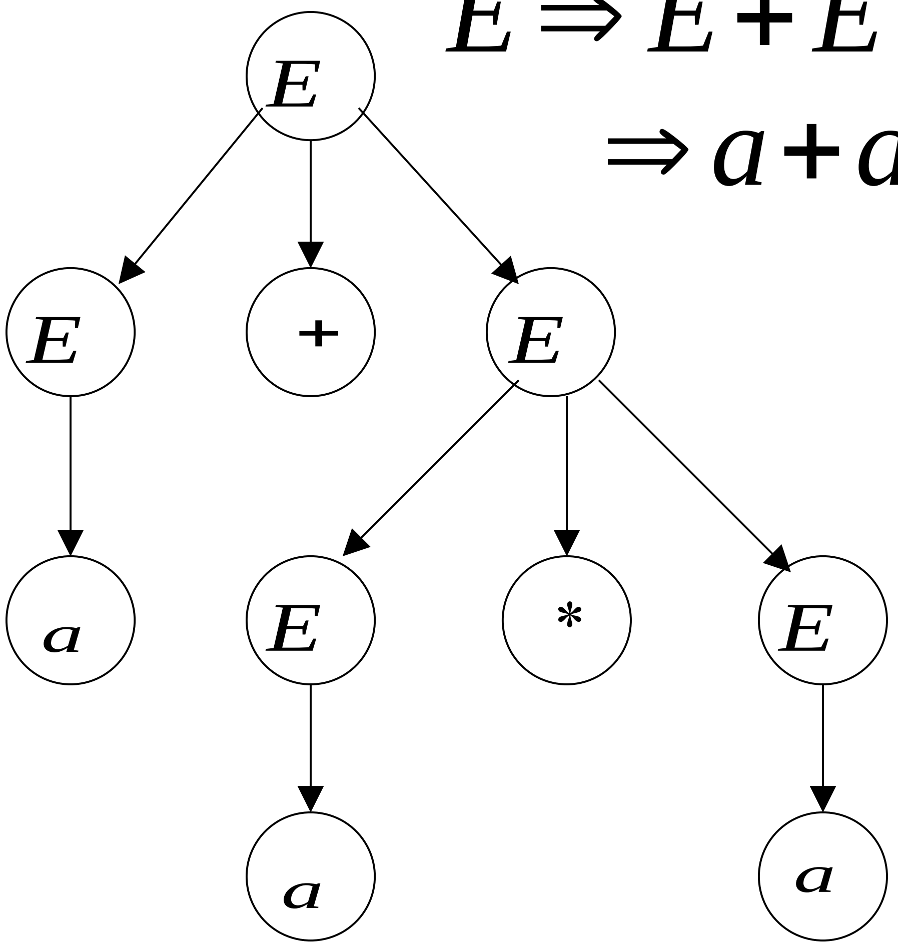
Example strings:

$$(a+a)*a+(a+a*(a+a))$$

Denotes any number

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

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

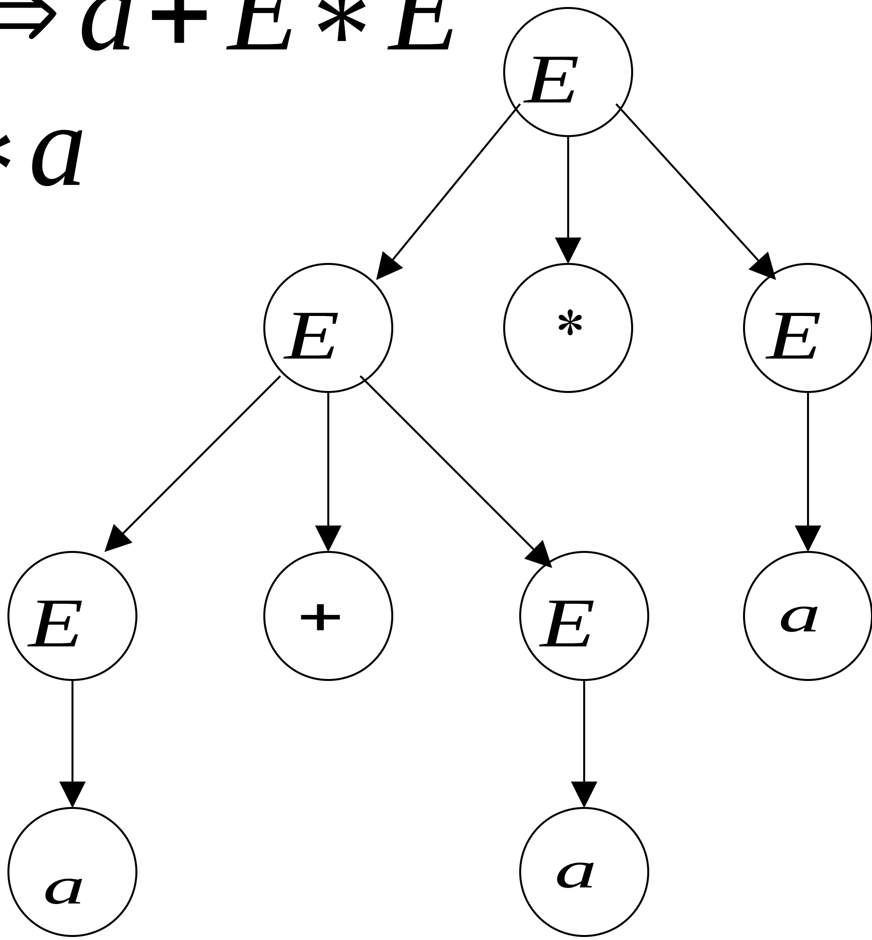


A leftmost derivation for $a + a * a$

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

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

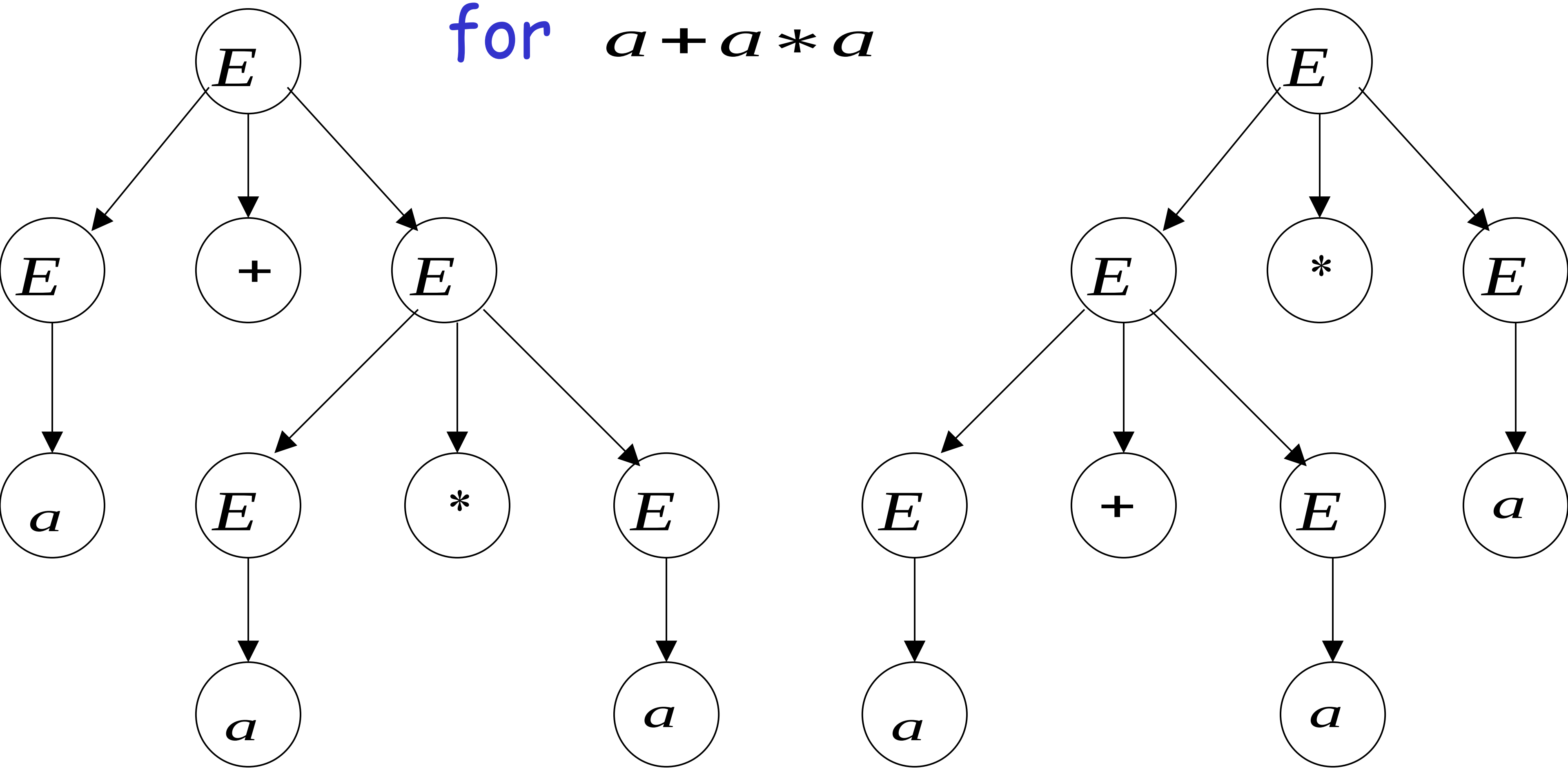
Another leftmost derivation for $a+a*a$



$$E \rightarrow E + E \;|\; E * E \;|\; (E) \;|\; a$$

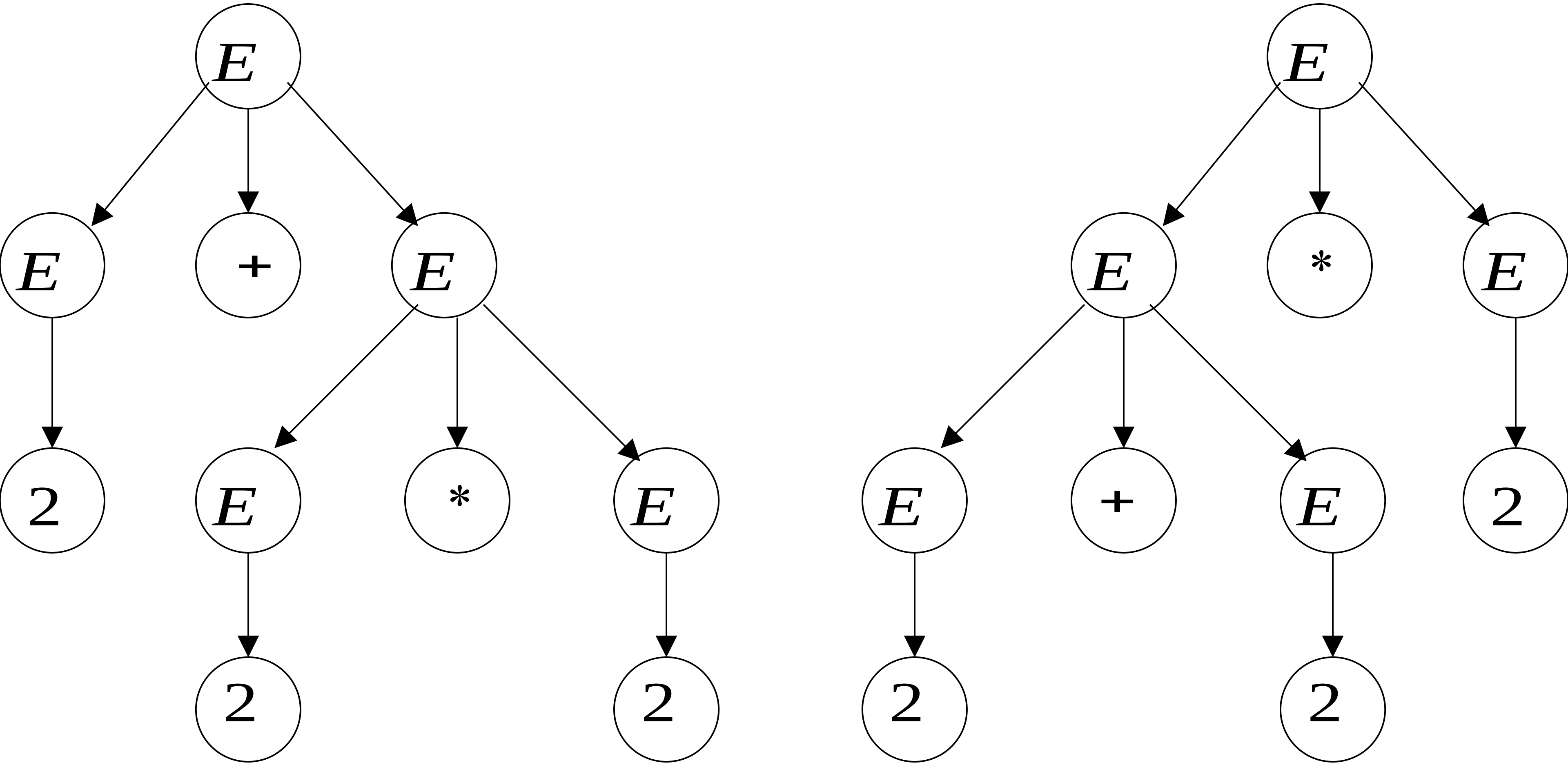
Two derivation trees

for $a + a * a$



take $a=2$

$$a+a*a=2+2*2$$

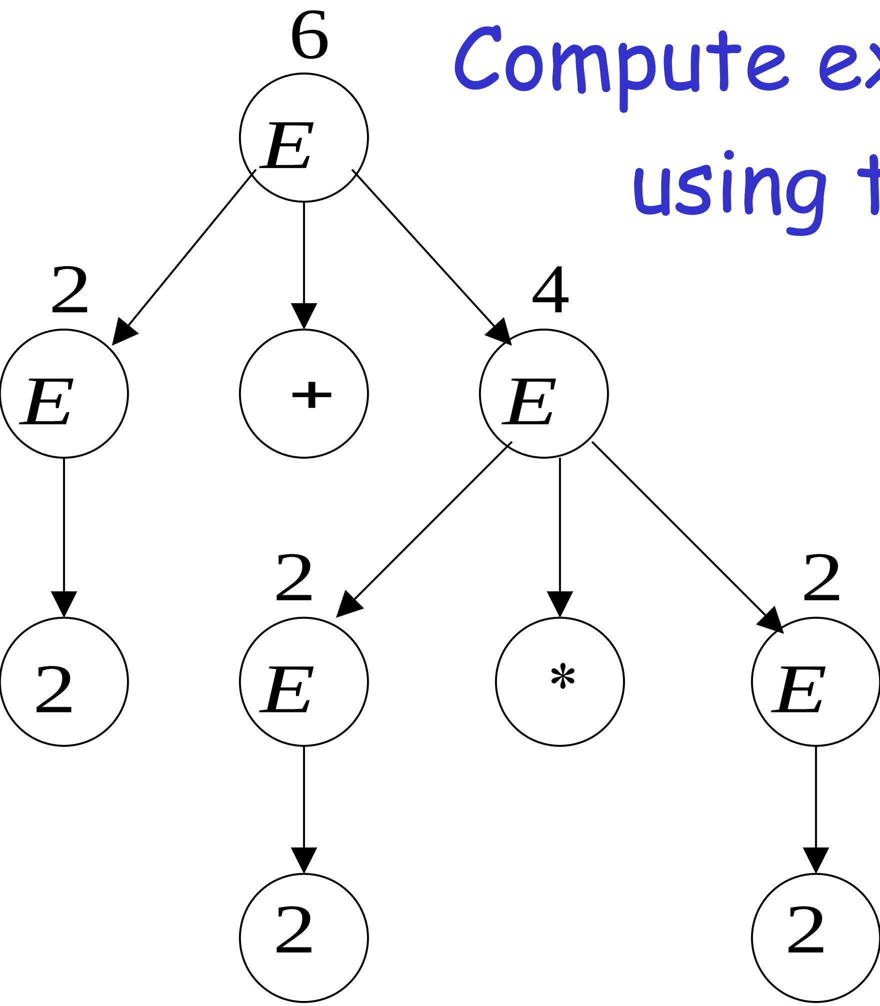


Good Tree

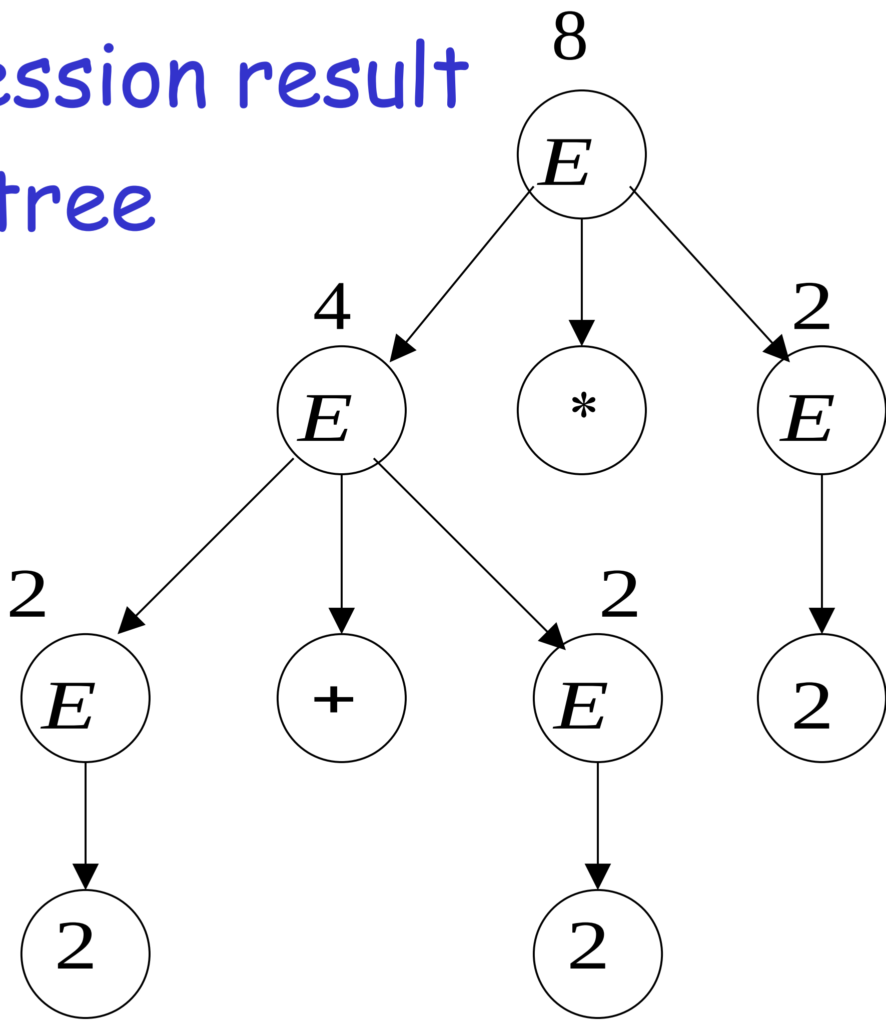
$2+2*2=6$

Bad Tree

$2+2*2=8$



Compute expression result using the tree



Two different derivation trees
may cause problems in applications which
use the derivation trees:

- Evaluating expressions

- In general, in compilers
  for programming languages

Ambiguous Grammar:

A context-free grammar G is ambiguous if there is a string $w \in L(G)$ which has:

two different derivation trees

or

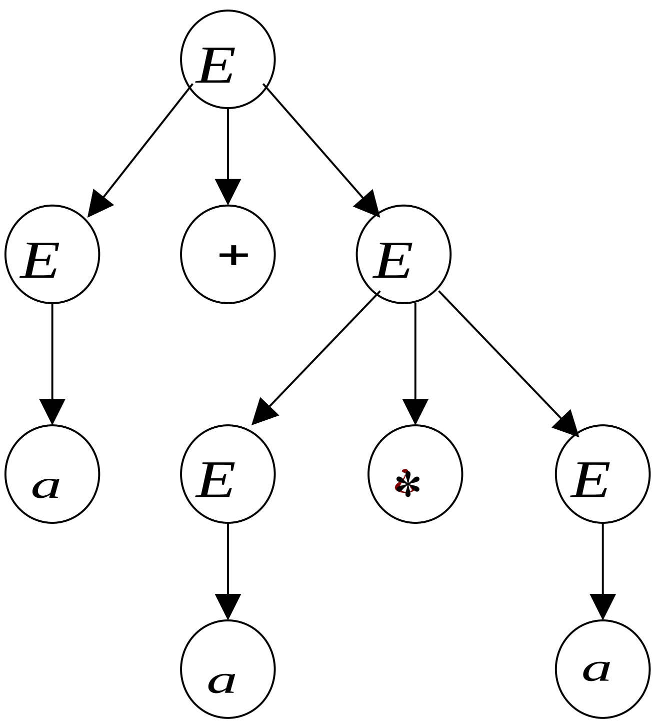
two leftmost derivations

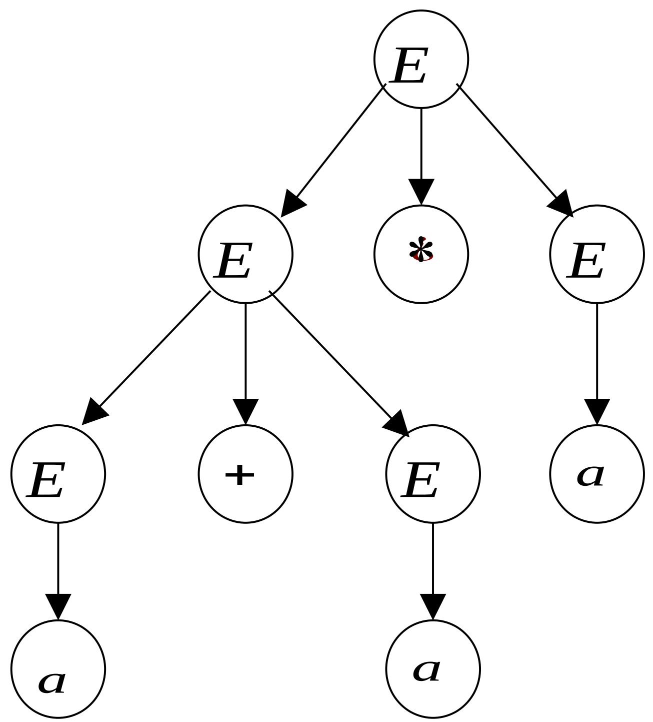
(Two different derivation trees give two different leftmost derivations and vice-versa)

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

this grammar is ambiguous since

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





$$E \rightarrow E + E \;|\; E * E \;|\; (E) \;|\; a$$

this grammar is ambiguous also because

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

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

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

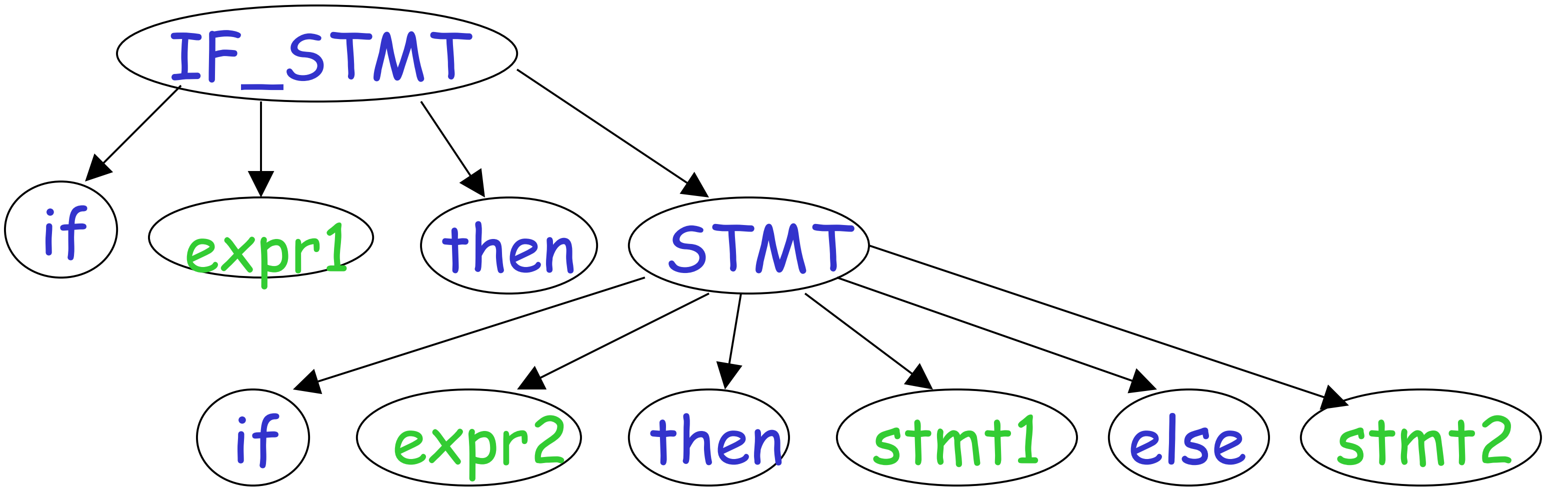
Another ambiguous grammar:

IF_STMT → if EXPR then STMT
| if EXPR then STMT else STMT

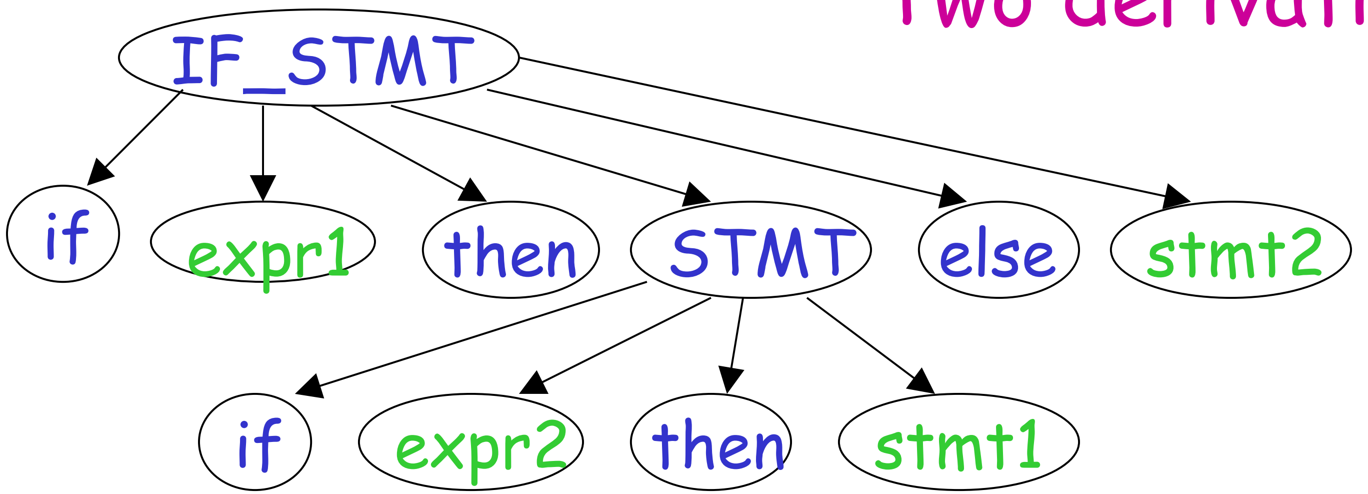
Variables   Terminals

Very common piece of grammar
in programming languages

If expr1 then if expr2 then stmt1 else stmt2



Two derivation trees



In general, ambiguity is bad
and we want to remove it

Sometimes it is possible to find
a non-ambiguous grammar for a language

But, in general we cannot do so

A successful example:

Ambiguous Grammar

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

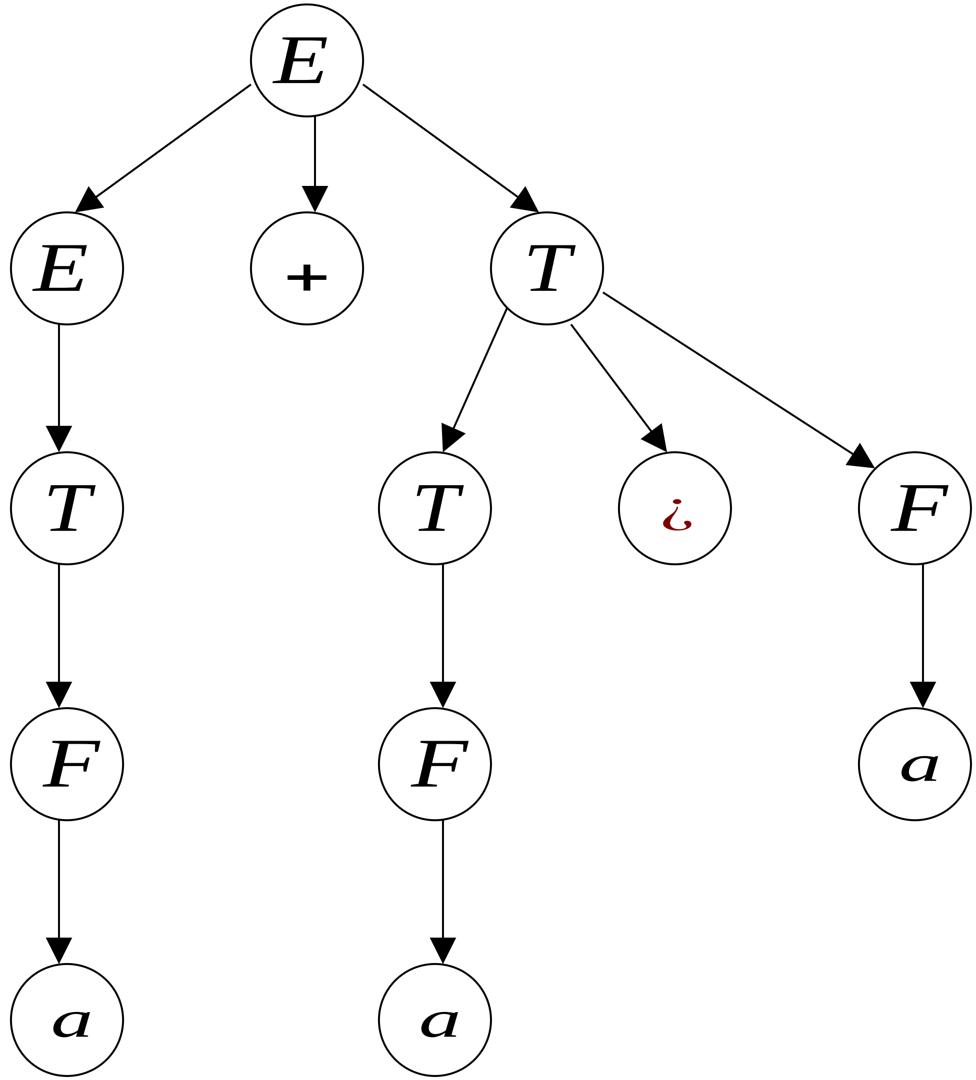
Equivalent Non-Ambiguous Grammar

$$E \to E + T \mid T$$
$$T \to T * F \mid F$$
$$F \to (E) \mid a$$

generates the same language

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

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



Unique derivation tree for $a+a*a$

An un-successful example:

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

$$n, m \geq 0$$

L is inherently ambiguous:

every grammar that generates this language is ambiguous

Example (ambiguous) grammar for L:

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

$$S \rightarrow S_1 \mid S_2$$

$$S_1 \rightarrow S_1 c \mid A$$
$$A \rightarrow aAb \mid \lambda$$

$$S_2 \rightarrow aS_2 \mid B$$
$$B \rightarrow bBc \mid \lambda$$

The string $a^n b^n c^n \in L$

has always two different derivation trees

(for any grammar)

For example