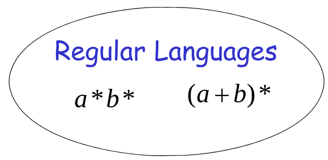
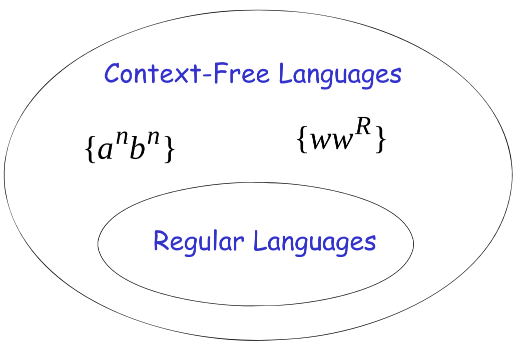
Context-Free Languages

class 8

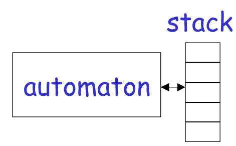
$\{a^n b^n : n \geq 0\}$ $\{ww^R\}$





Context-Free Languages

Context-Free Grammars ↔ Pushdown Automata



Context-Free Grammars

Example

A context-free grammar G:

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

A derivation:

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

A context-free grammar G:

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

Another derivation:

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

7

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

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

Describes parentheses: (((( ))))

8

Example

A context-free grammar G:

$$S \to aSa$$
$$S \to bSb$$
$$S \to \lambda$$

A derivation:

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

9

A context-free grammar G:

$$S \to aSa$$
$$S \to bSb$$
$$S \to \lambda$$

Another derivation:

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

10

$$S \to aSa$$
$$S \to bSb$$
$$S \to \lambda$$

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

11

Example

A context-free grammar G:

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

A derivation:

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

12

A context-free grammar G:
$$S \to aSb$$
$$S \to SS$$
$$S \to \lambda$$

A derivation:

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

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

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

Describes matched parentheses: () ((( ))) (( ))

Definition: Context-Free Grammars

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

Variables

Terminal symbols

Start variable

Productions of the form:

$$A \to x$$

Variable

String of variables and terminals

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

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

Definition: Context-Free Languages

A language L is context-free

if and only if

there is a context-free grammar G with $L = L(G)$

Derivation Order

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

Leftmost derivation:

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

Rightmost derivation:

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

$$S \to aAB$$
$$A \to bBb$$
$$B \to A \mid \lambda$$

Leftmost derivation:

$$S \Rightarrow aAB \Rightarrow abBbB \Rightarrow abAbB \Rightarrow abbBbbB$$
$$\Rightarrow abbbbB \Rightarrow abbbb$$

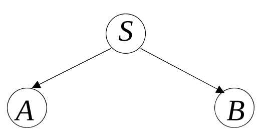
Rightmost derivation:

$$S \Rightarrow aAB \Rightarrow aA \Rightarrow abBb \Rightarrow abAb$$
$$\Rightarrow abbBbb \Rightarrow abbbb$$

Derivation Trees

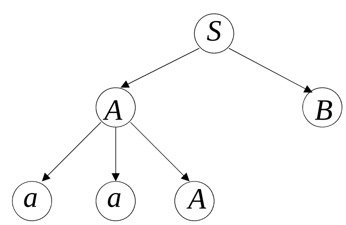
$S \to AB$ $\quad A \to aaA \mid \lambda$ $\quad B \to Bb \mid \lambda$

$$S \Rightarrow AB$$



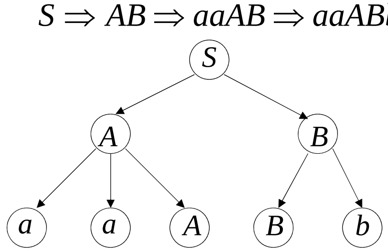
$S \to AB$ $\quad A \to aaA \mid \lambda$ $\quad B \to Bb \mid \lambda$

$$S \Rightarrow AB \Rightarrow aaAB$$



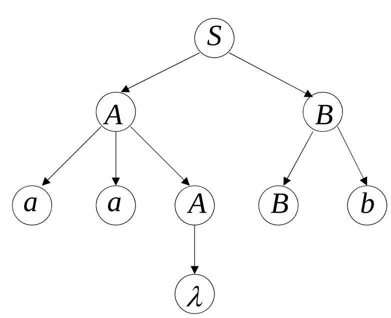
$S \to AB$ $\quad A \to aaA \mid \lambda$ $\quad B \to Bb \mid \lambda$

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



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

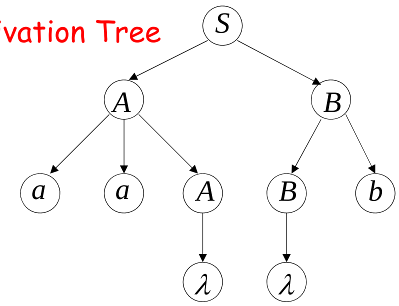
$$S \Rightarrow AB \Rightarrow aaAB \Rightarrow aaABb \Rightarrow aaBb$$



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

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

Derivation Tree

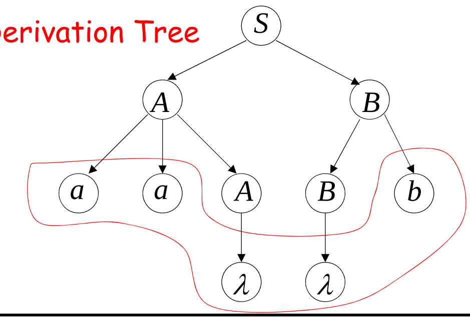


25

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

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

Derivation Tree



yield

$aa\lambda\lambda b$

$= aab$

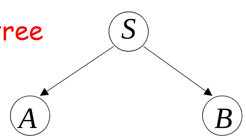
26

Partial Derivation Trees

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

$S \Rightarrow AB$

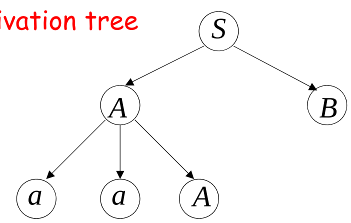
Partial derivation tree



27

$S \Rightarrow AB \Rightarrow aaAB$

Partial derivation tree

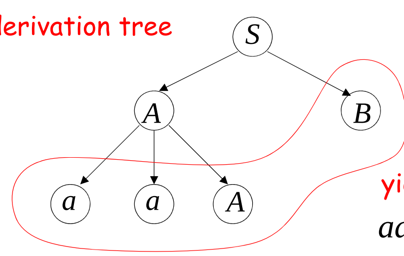


28

$S \Rightarrow AB \Rightarrow aaAB$

sentential form

Partial derivation tree



yield

$aaAB$

29

Sometimes, derivation order doesn't matter

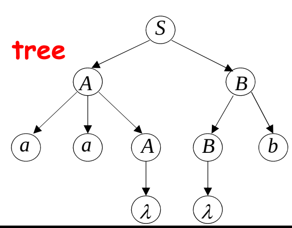
Leftmost:

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

Rightmost:

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

Same derivation tree

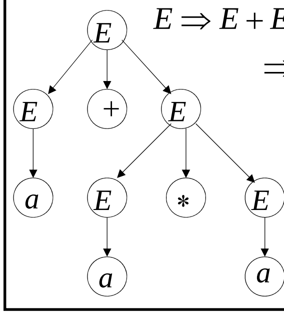


30

Ambiguity

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

$$a+a*a$$



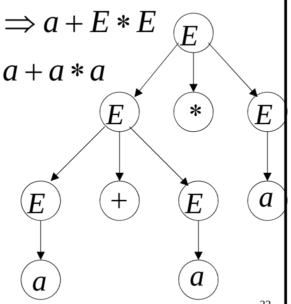
$$E \Rightarrow E+E \Rightarrow a+E \Rightarrow a+E*E$$
$$\Rightarrow a+a*E \Rightarrow a+a*a$$

leftmost derivation

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

$$a+a*a$$

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

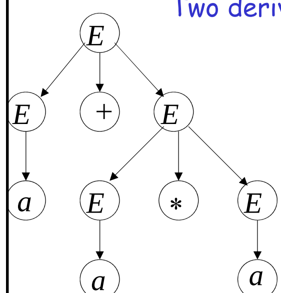


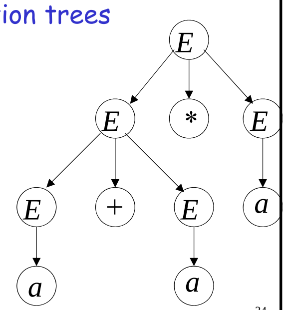
leftmost derivation

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

$$a+a*a$$

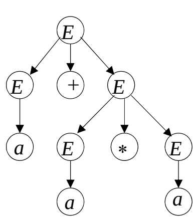
Two derivation trees

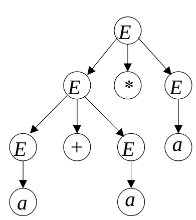




The grammar $E \to E+E \;\mid\; E*E \mid (E) \mid a$ is **ambiguous:**

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





The grammar $E \to E+E \;\mid\; E*E \mid (E) \mid a$ is **ambiguous:**

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 G is **ambiguous**

if some string $w \in L(G)$ has:

two or more derivation trees

In other words:

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

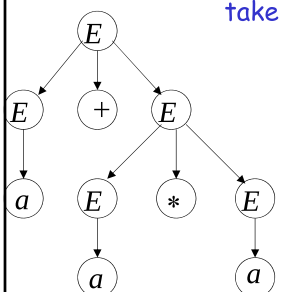
if some string $w \in L(G)$ has:

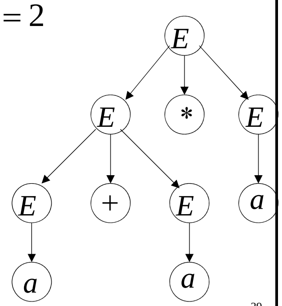
two or more leftmost derivations
(or rightmost)

Why do we care about ambiguity?

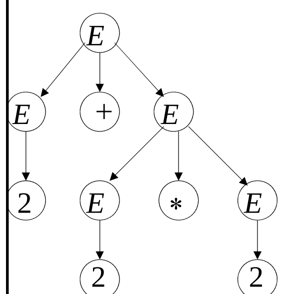
$$a + a * a$$

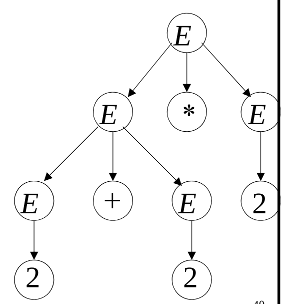
take $a = 2$





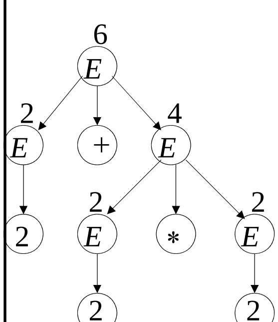
$$2 + 2 * 2$$

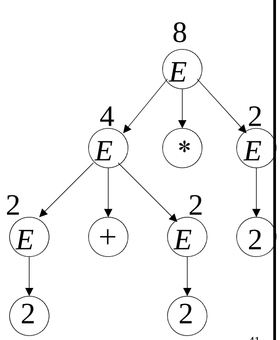




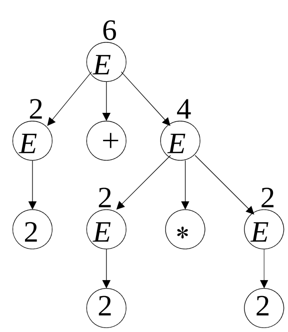
$2 + 2 * 2 = 6$

$2 + 2 * 2 = 8$





Correct result: $2 + 2 * 2 = 6$



- Ambiguity is **bad** for programming languages

- We want to remove ambiguity

We fix the ambiguous grammar:

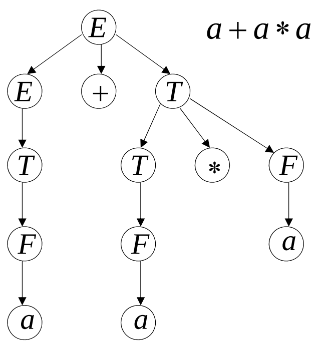
$$E \to E+E \ \mid E*E \mid (E) \mid a$$

New non-ambiguous grammar:

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

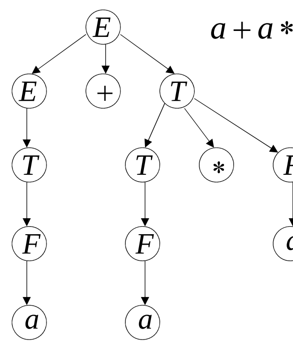
$$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T \Rightarrow a+T \Rightarrow a+T*F$$
$$\Rightarrow a+F*F \Rightarrow a+a*F \Rightarrow a+a*a$$

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



$a+a*a$

Unique derivation tree



$a+a*a$

The grammar G:

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

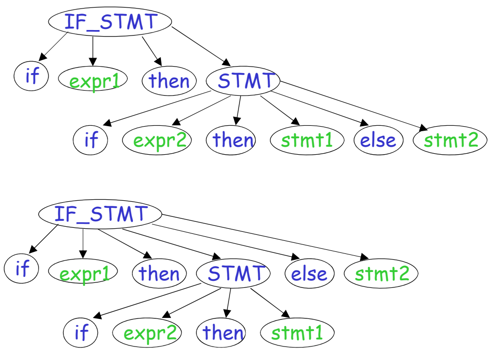
is non-ambiguous:

Every string $w \in L(G)$ has a unique derivation tree

Another Ambiguous Grammar

IF_STMT $\to$ if EXPR then STMT
$\mid$ if EXPR then STMT else STMT

If expr1 then if expr2 then stmt1 else stmt2



49

Inherent Ambiguity

Some context free languages
have only ambiguous grammars

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

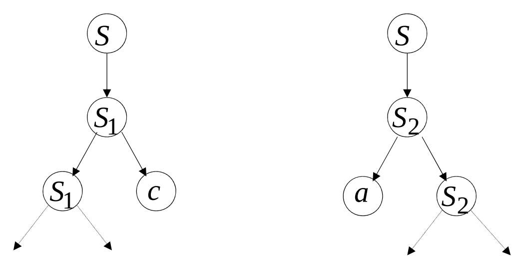
$S \to S_1 \mid S_2$ $\quad S_1 \to S_1 c \mid A$ $\quad S_2 \to aS_2 \mid B$

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

50

The string $a^n b^n c^n$

has two derivation trees



51

Compilers

52

Machine Code

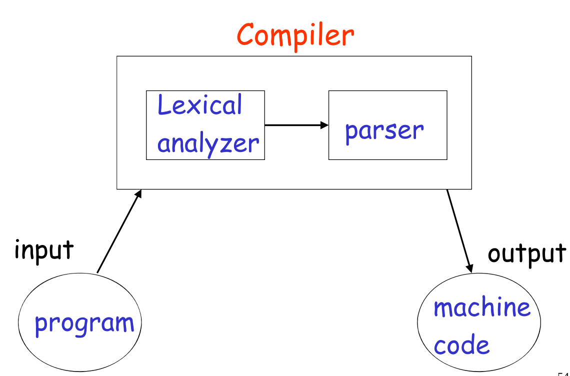
Program

```
v = 5;
if (v>5)
  x = 12 + v;
while (x !=3) {
  x = x - 3;
  v = 10;
}
......
```

Compiler

```
Add v,v,0
cmp v,5
jmplt ELSE
THEN:
  add x, 12,v
ELSE:
WHILE:
cmp x,3
...
```

53



54

A parser knows the grammar of the programming language

55

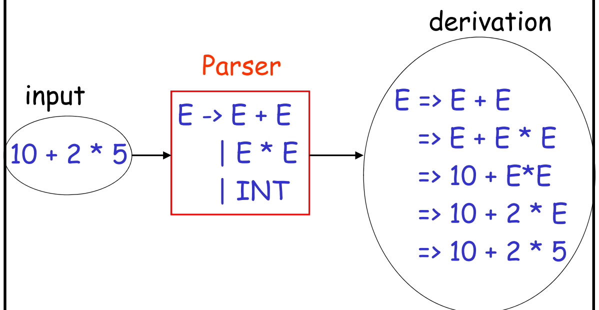
Parser

PROGRAM → STMT_LIST
STMT_LIST → STMT; STMT_LIST | STMT;
STMT → EXPR | IF_STMT | WHILE_STMT
| { STMT_LIST }

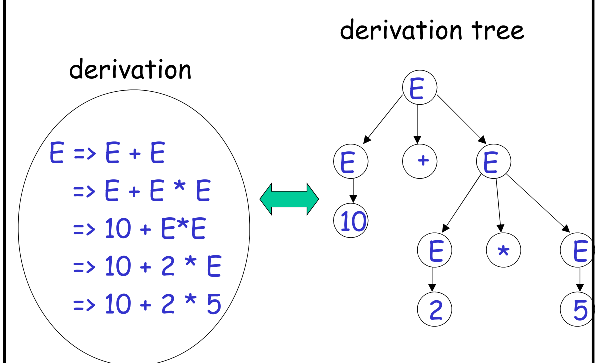
EXPR → EXPR + EXPR | EXPR - EXPR | ID
IF_STMT → if (EXPR) then STMT
| if (EXPR) then STMT else STMT
WHILE_STMT → while (EXPR) do STMT

56

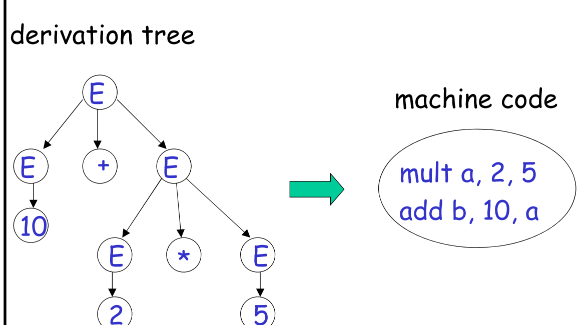
The parser finds the derivation of a particular input



57



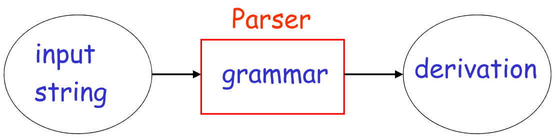
58



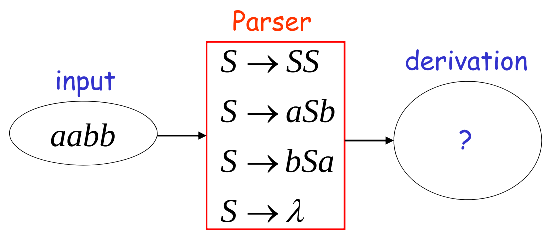
59

Parsing

60



Example:



Exhaustive Search

$$S \to SS \mid aSb \mid bSa \mid \lambda$$

Phase 1:

$S \Rightarrow SS$

$S \Rightarrow aSb$

$S \Rightarrow bSa$

$S \Rightarrow \lambda$

Find derivation of $aabb$

All possible derivations of length 1

$S \Rightarrow SS$

$S \Rightarrow aSb$

$\cancel{S \Rightarrow bSa}$

$\cancel{S \Rightarrow \lambda}$

$aabb$

Phase 2 $\quad S \to SS \mid aSb \mid bSa \mid \lambda$

Phase 1

$S \Rightarrow SS$

$S \Rightarrow SS \Rightarrow SSS$

$S \Rightarrow SS \Rightarrow aSbS$

$\cancel{S \Rightarrow SS \Rightarrow bSaS}$

$S \Rightarrow SS \Rightarrow S$

$aabb$

$S \Rightarrow aSb$

$S \Rightarrow aSb \Rightarrow aSSb$

$S \Rightarrow aSb \Rightarrow aaSbb$

$\cancel{S \Rightarrow aSb \Rightarrow abSab}$

$\cancel{S \Rightarrow aSb \Rightarrow ab}$

Phase 2 $\quad S \to SS \mid aSb \mid bSa \mid \lambda$

$S \Rightarrow SS \Rightarrow SSS$

$S \Rightarrow SS \Rightarrow aSbS$

$S \Rightarrow SS \Rightarrow S$

$aabb$

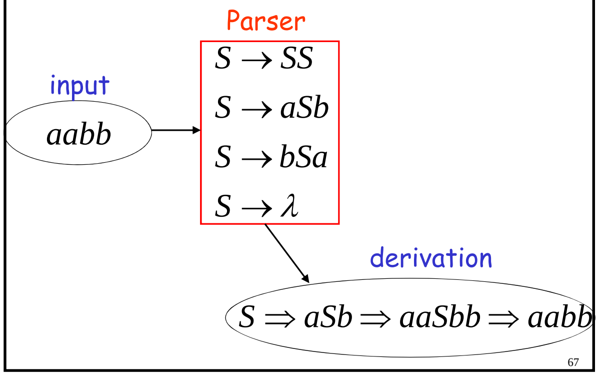
$S \Rightarrow aSb \Rightarrow aSSb$

$S \Rightarrow aSb \Rightarrow aaSbb$

Phase 3

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

Final result of exhaustive search (top-down parsing)



67

Time complexity of exhaustive search

Suppose there are no productions of the form

$$A \to \lambda$$

$$A \to B$$

Number of phases for string w : $2|w|$

68

For grammar with k rules

Time for phase 1: k

k possible derivations

69

Time for phase 2: k^2

k^2 possible derivations

70

Time for phase $2|w|$: $k^{2|w|}$

$k^{2|w|}$ possible derivations

71

Total time needed for string w:

$$k + k^2 + \cdots + k^{2|w|}$$

phase 1 phase 2 phase 2|w|

Extremely bad!!!

72

There exist faster algorithms
for specialized grammars

S-grammar: $A \to ax$

symbol

string
of variables

Pair (A, a) appears once

73

S-grammar example:

$$S \to aS$$
$$S \to bSS$$
$$S \to c$$

Each string has a unique derivation

$$S \Rightarrow aS \Rightarrow abSS \Rightarrow abcS \Rightarrow abcc$$

74

For S-grammars:

In the exhaustive search parsing
there is only one choice in each phase

Time for a phase: 1

Total time for parsing string w: $|w|$

75

For general context-free grammars:

There exists a parsing algorithm
that parses a string $|w|$
in time $|w|^3$

(we will show it in the next class)

76