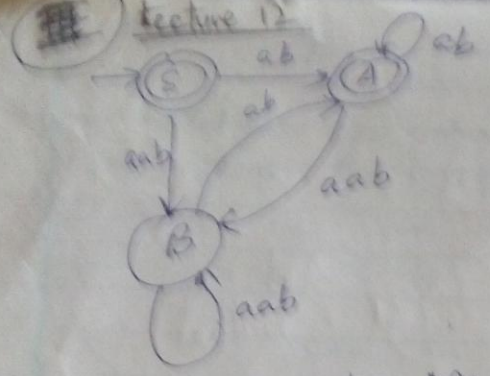
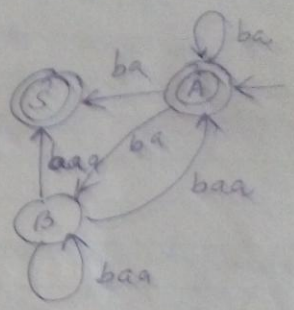
Lecture - 12

12/05/20

Lecture 12



obtain LLG (1)



Step 1 :→ Reverse the DFA

Step 2 obtain RLG for reversed DFA

$A \rightarrow baS \mid baB \mid baA \mid \varepsilon$

$B \rightarrow baaS \mid baaA \mid baaB$

$S \rightarrow \varepsilon$

Step 3 Reverse the productions.

RLG

$A \rightarrow baA \mid baB \mid baS \mid \varepsilon$

$B \rightarrow baaB \mid baaA \mid baaS$

$S \rightarrow \varepsilon$

$G_1 = (V, T, P, S)$

LLG

$A \rightarrow Aab \mid Bab \mid Sab \mid \varepsilon$

$B \rightarrow Baab \mid Aaab \mid Saab$

$S \rightarrow \varepsilon$

test a string

aababaabab

on both Gs

Obtain a Gram to generate an AE using the operators +, −, *, /, ^ (power) (2)
An identifier can start with any of the letters from {a, b, c} and can be followed by zero or more symbols from {a, b, c}.

Sol: An AE can be recursively defined as

(i) An expression E can be an identifier.

(ii) If E is any AE, then

(1) E+E (3) E*E (5) E^E

(2) E−E (4) E/E (6) (E)

are all AE's

An identifier I of length at least one can be generated using any combinations of a's, b's and c's using the following productions

I = Ia | Ib | Ic | a | b | c

(1) Base Case

E → I

(ii) by 2nd def.

$E \to E * E$

$E \to E - E$

$E \to E * E$

$E \to E / E$

$E \to E \hat{} E$

$E \to (E)$

So the complete grammar to generate an AE is given by

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

$V = \{E, I\}$

$T = \{+, -, *, /, \hat{}, a, b, c\}$

$P = \{$

$E \to$ all the above

$I \to Ia \mid Ib \mid Ic \mid a \mid b \mid c$

$\}$

S is the start symbol.

$E \to I$

$E \to E+E \mid E-E \mid E*E \mid E/E \mid E\hat{}E \mid (E)$

$I \to Ia \mid Ib \mid Ic \mid a \mid b \mid c$

Leftmost & Rightmost Derivations (4)

(a) Derivations of CFGs are of two types:
(i) leftmost (ii) rightmost

(b) It is based on the variable replacement in derivation.

(c) If leftmost variable is replaced on the RHS of a production ⇒ leftmost derivation

(d) If rightmost variable is replaced on the RHS of the production ⇒ Rightmost derivation.

~~E → E+E~~

obtain id+ id*id

Leftmost

E → E+E
→ id+E
→ id+ E*E
→ id+ id*E
→ id+ id*id

right most

E → E*E
→ E*id
→ E+E*id
→ E+id*id
→ id+id*id

Problem: Leftmost derivation for the string aaabbabbba using the following grammar (5)

$S \rightarrow aB \mid bA$

$A \rightarrow aS \mid bAA \mid a$

$B \rightarrow bS \mid aBB \mid b$

Sol

The leftmost derivation for the string is

$S \rightarrow aB$ using $S \rightarrow aB$

$\rightarrow aaBB$ using $B \rightarrow aBB$

$\rightarrow aaaBBB$ using $B \rightarrow aBB$

$\rightarrow aaabBB$ using $B \rightarrow b$.

$\rightarrow aaabbB$ using $B \rightarrow b$

$\rightarrow aaabbaBB$ using $B \rightarrow aBB$

$\rightarrow aaabbabB$ using $B \rightarrow b$.

$\rightarrow aaabbabbS$ using $B \rightarrow bS$

$\rightarrow aaabbabbbA$ using $S \rightarrow bA$

$\rightarrow aaabbabbba$ " $S \rightarrow a$

---

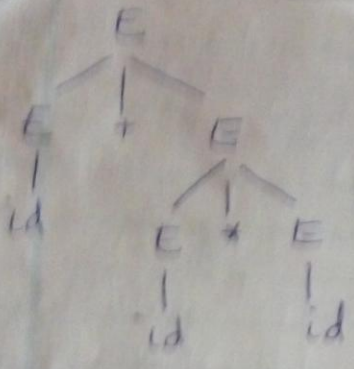
repeat for rightmost derivation

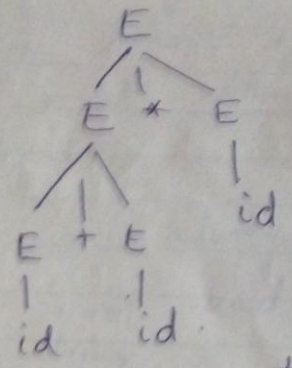
Derivation Trees (Parse trees)

- Derivations can be shown in the form of a tree
- Such trees are called derivation trees or parse trees
- Leftmost and rightmost derivation trees exist

Def: Let $G = (V, T, P, S)$ be a CFG. The tree is a derivation tree with the following properties:

1. The root has a label S
2. Every vertex has a label which is in $(V \cup T \cup \varepsilon)$
3. Every leaf node has label from T and an interior vertex has a label from V.
4. If a vertex is labelled A and if $X_1, X_2, X_3 \ldots X_n$ are its children from left (to right), then $A \to X_1 X_2 X_3 \ldots X_n$ must be a production in P.





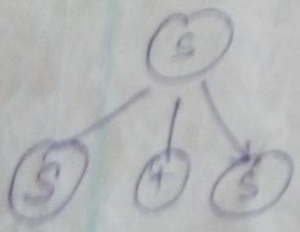
Derivation tree

(id + id * id) → yield of the tree

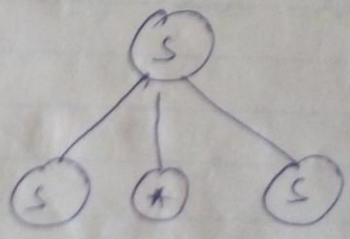
Derivation tree

CFG $S \to S+S \mid S*S \mid a \mid b$

Represent each by a derivation tree

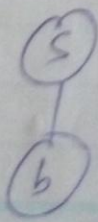


$S \to S+S$



$S \to S*S$

$S \to a$



$S \to b$

(7) Yield of a tree is the string obtained by only reading the leaves of the tree from left to right without considering the ε-symbols. (8)

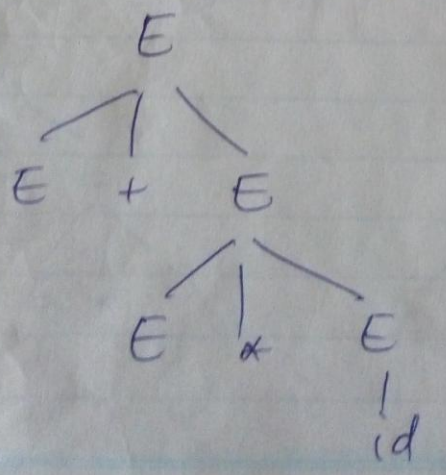
(8) Yield of the tree is always derived from the root and is always a terminal string

---

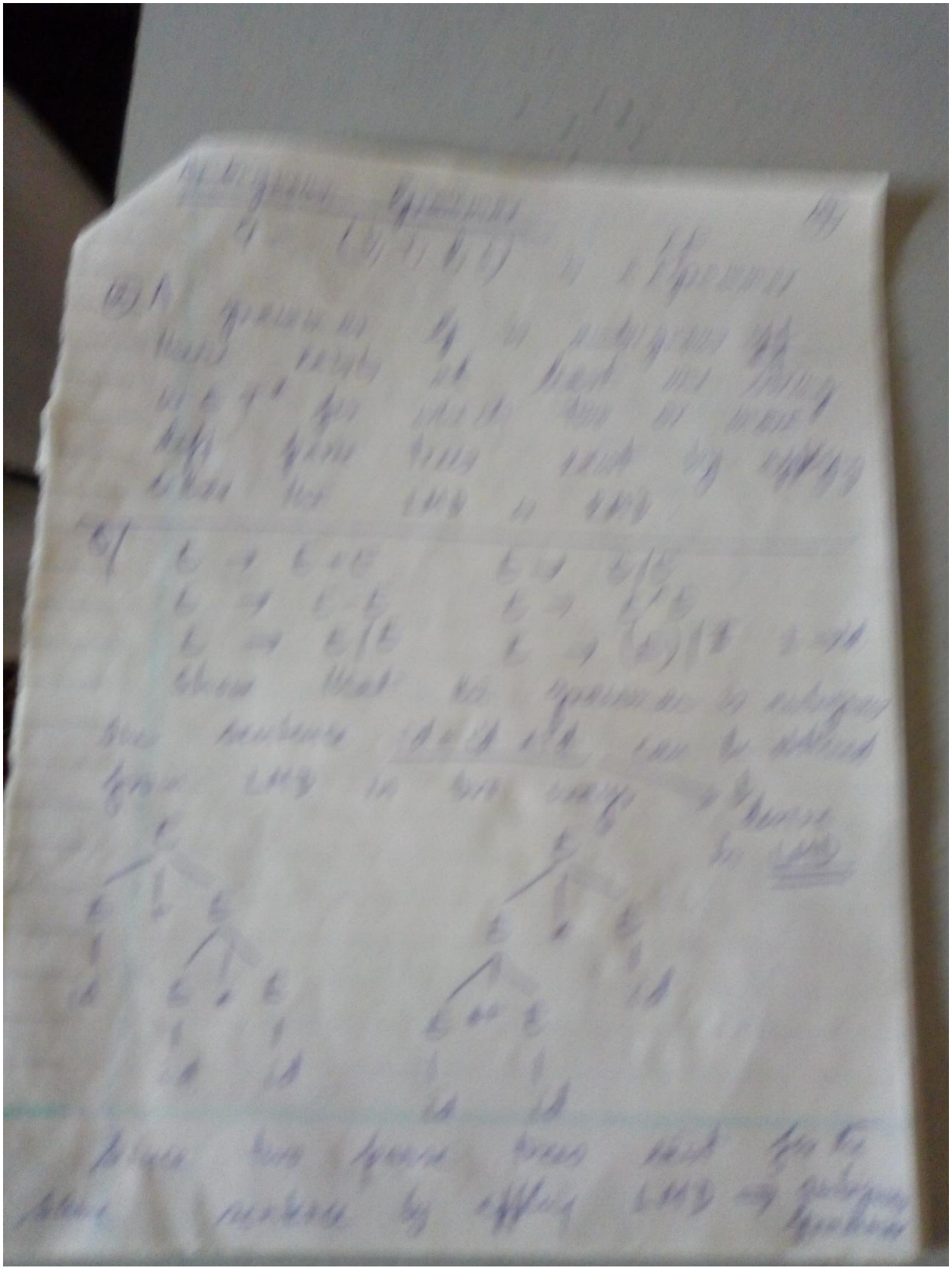
Partial Parse tree / Partial Derivation tree

Same as Parse / derivation ~~tree~~ except

(3) Every leaf node has label from $(V \cup T \cup \varepsilon)$



Partial Parse tree



Is the following grammar ambiguous (10)

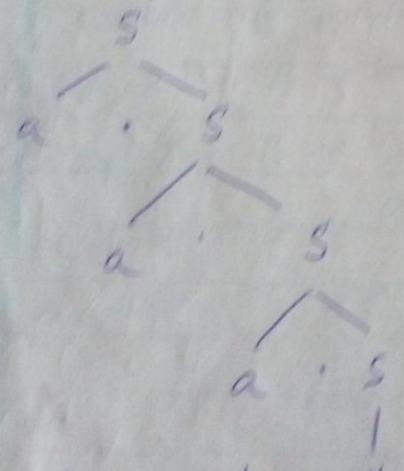
$S \to aS \mid X$

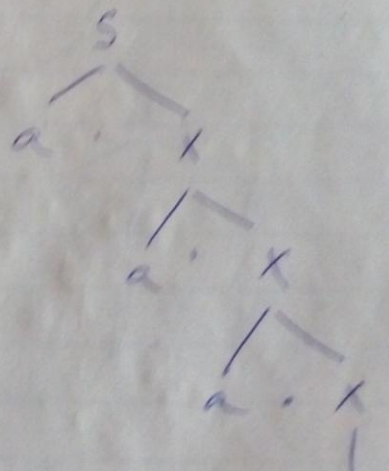
$X \to aX \mid a$

Sol Consider two leftmost derivations for the string aaaa:

$S \to aS$
$\to aaS$
$\Rightarrow aaaS$
$\Rightarrow aaaX$
$\Rightarrow aaaa$

$S \to aX$
$\Rightarrow aaX$
$\Rightarrow aaaX$
$\Rightarrow aaaa$





Two PTs exist
∴ ambiguous

Is the following grammar ambiguous? (11)

$S \to aB \mid bA$

$A \to aS \mid bAA \mid a$

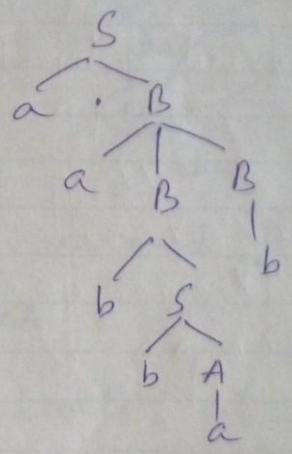
$B \to bS \mid aBB \mid b$

Sol

Consider LMD.

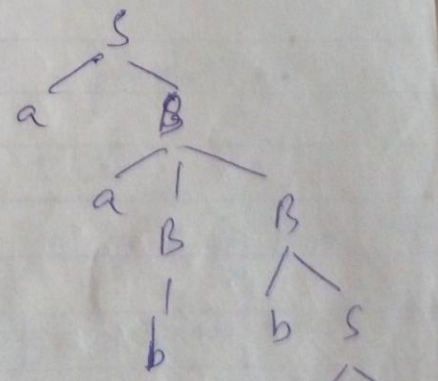
I

$S \to aB$	$S \to aB$
$\to aaBB$	$B \to aBB$
$\to aabSB$	$B \to bS$
$\to aabbAB$	$S \to bA$
$\to aabbaB$	$A \to a$
$\to aabbab$	$B \to b$



II

$S \to aB$	$S \to aB$
$S \to aaBB$	$B \to aBB$
$S \to aabB$	$B \to b$
$S \to aabbS$	$B \to bS$
$S \to aabbaB$	$S \to aB$
$S \to aabbab$	$B \to b$



$\Rightarrow$ Two PTs exist for same string $\Rightarrow$ grammar is ambiguous