CSE211 – Formal Languages and Automata Theory

U2L3 – Derivation and Parse Tree

Dr. P. Saravanan
School of Computing
SASTRA Deemed University

Agenda

- Derivation using Grammar
- Types of derivation
- Examples
- Comparison of LM and RM derivation
- Definition of CFL
- Sentential Forms
- Parse Tree
- Examples for parse tree
- Yield of a parse tree

Types of Derivations

- **Definitions:**
- *Leftmost derivation:* *R*eplacing the leftmost variable in each derivation step (represented by the notation or, for typing convenience, also by $\Rightarrow_{lm}$)
- *Rightmost derivation:* *R*eplacing the rightmost variable in each derivation step (represented by $\underset{rm}{\Rightarrow}$ or by $\Rightarrow_{rm}$)

Comparison of LM &RM

Leftmost Derivation of $a*(a+b00)$

$$
\begin{array}{llll}
E & \Rightarrow_{lm} E*E & & \text{apply (3)} \\
 & \Rightarrow_{lm} I*E & & \text{apply (1)} \\
 & \Rightarrow_{lm} a*E & & \text{apply (5)} \\
 & \Rightarrow_{lm} a*(E) & & \text{apply (4)} \\
 & \Rightarrow_{lm} a*(E+E) & & \text{apply (2)} \\
 & \Rightarrow_{lm} a*(I+E) & & \text{apply (1)} \\
 & \Rightarrow_{lm} a*(a+E) & & \text{apply (5)} \\
 & \Rightarrow_{lm} a*(a+I) & & \text{apply (1)} \\
 & \Rightarrow_{lm} a*(a+I0) & & \text{apply (9)} \\
 & \Rightarrow_{lm} a*(a+b00) & & \text{apply (9)} \\
 & \Rightarrow_{lm} a*(a+b00) & & \text{apply (6)}
\end{array}
$$

Rightmost Derivation of $a*(a+b00)$

$$
\begin{array}{llll}
E & \Rightarrow_{rm} E*E & & \text{apply (3)} \\
 & \Rightarrow_{rm} E*(E) & & \text{apply (4)} \\
 & \Rightarrow_{rm} E*(E+E) & & \text{apply (2)} \\
 & \Rightarrow_{rm} E*(E+I) & & \text{apply (1)} \\
 & \Rightarrow_{rm} E*(E+I0) & & \text{apply (9)} \\
 & \Rightarrow_{rm} E*(E+I00) & & \text{apply (9)} \\
 & \Rightarrow_{rm} E*(E+b00) & & \text{apply (6)} \\
 & \Rightarrow_{rm} E*(I+b00) & & \text{apply (1)} \\
 & \Rightarrow_{rm} E*(a+b00) & & \text{apply (5)} \\
 & \Rightarrow_{rm} I*(a+b00) & & \text{apply (1)} \\
 & \Rightarrow_{rm} a*(a+b00) & & \text{apply (5)}
\end{array}
$$

Sentential forms

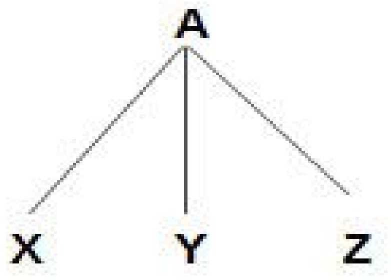
- Derivations from the start symbol are called *sentential forms*.
  - Given a CFG $G = (V, T, P, S)$, if $S \overset{*}{\Rightarrow} \alpha$ with $\alpha \in (V \cup T)^*$, then α is a sentential form.
  - If $S \underset{lm}{\overset{*}{\Rightarrow}} \alpha$ where $\alpha \in (V \cup T)^*$, then α is a left-sentential form.
  - If $S \underset{rm}{\overset{*}{\Rightarrow}} \alpha$ where $\alpha \in (V \cup T)^*$, then α is a right-sentential form.

Parse Trees

- **Parse tree** is a hierarchical structure which represents the derivation of the grammar to yield input strings
- Root node of **parse tree** has the start symbol of the given grammar from where the derivation proceeds
- Leaves of parse tree represent terminals
- Each interior node represents productions of grammar
- Example:

  If *A -> xyz* is a production, then

  the parse tree will have *A* as root node whose children are *x*, *y* and *z* from its left to right.



Advantages of parse trees

- In a compiler, the parse tree structure facilitates translation of the source program into recursive executable codes.
- Parse trees are closely related to derivations and recursive inferences
- An important application of the parse tree is the study of grammatical ambiguity which makes the grammar unsuitable for a programming language

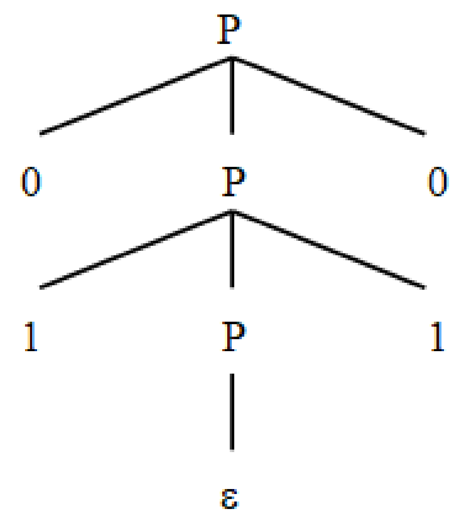
Constructing Parse Trees

- **Definition:** Given a grammar $G = (V, T, P, S)$, the parse tree is defined in the following way.
  - Each interior node is labeled by a variable in V
  - Each leaf is labeled by either a terminal, or ε
  - If ε is the label, it must be the only child of its parent
  - If an interior node is labeled A, and its children are labeled $X_1, X_2, \ldots, X_k$, respectively, from the left, then $A \rightarrow X_1X_2\ldots X_k$ is a production in P.

Construction of Parse Tree

- *Example 1:* Construct parse tree for the palindrome string 0110.
- The set of the five productions (1)~(5) below:
  - $P \rightarrow \varepsilon$, $P \rightarrow 0$, $P \rightarrow 1$, $P \rightarrow 0P0$, $P \rightarrow 1P1$
- Starting symbol P

The Parse Tree is



The Derivation is

P => 0P0 Use (4)
P => 01P10 apply (5)
P => 01ε10 apply (1)
P => 0110

The Yield of a Parse Tree

- The *yield* of a parse tree is the string obtained by concatenating all the leaves from the left, like

  01ε10= 0110 for the tree of the last example

- Showing the yields of the parse trees of a grammar *G* is another way to describe the language of *G*

Inference, Derivations, and Parse Trees

- **Theorems:** Given a grammar $G = (V, T, P, S)$, the following facts are all equivalent:
  - The recursive inference procedure determines that terminal string w is in the language of variable A;

$$A \overset{*}{\Rightarrow} w;$$

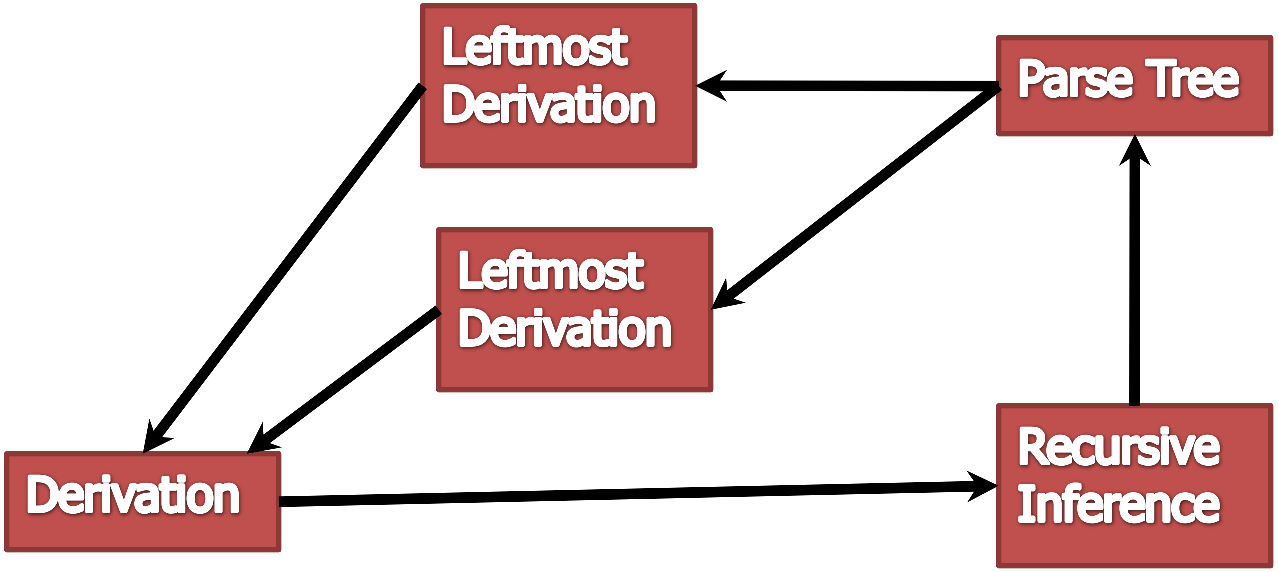
$$A \underset{lm}{\overset{*}{\Rightarrow}} w;$$

$$A \underset{rm}{\overset{*}{\Rightarrow}} w;$$

- There is a parse tree with root A and yield w.

Inference, Derivations, and Parse Trees

- Equivalences of ways to generate strings based on grammars



Parse Tree: Example

- Given the grammar:

  S → if (E) S | if (E) S else S

  S → other

  E → expr

- Draw the parse tree for

  if (expr) if (expr) other else other

Parse Tree Example

if

- if (expr) if (expr) other else other

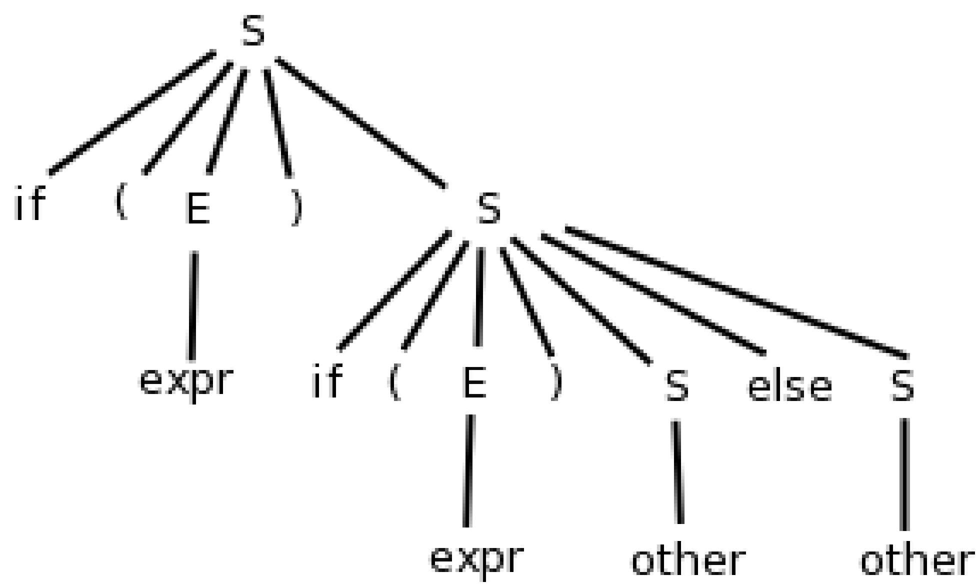
Production Rules

S → if (E) S / if (E) S else S
S → other
E → expr

Derivation 1:

S ⇒ if (E) S
⇒ if (E) if (E) S else S
⇒ if (E) if (E) S else other
⇒ if (E) if (E) other else other
⇒ if (E) if (expr) other else other
⇒ if (expr) if (expr) other else other

Parse Tree 1



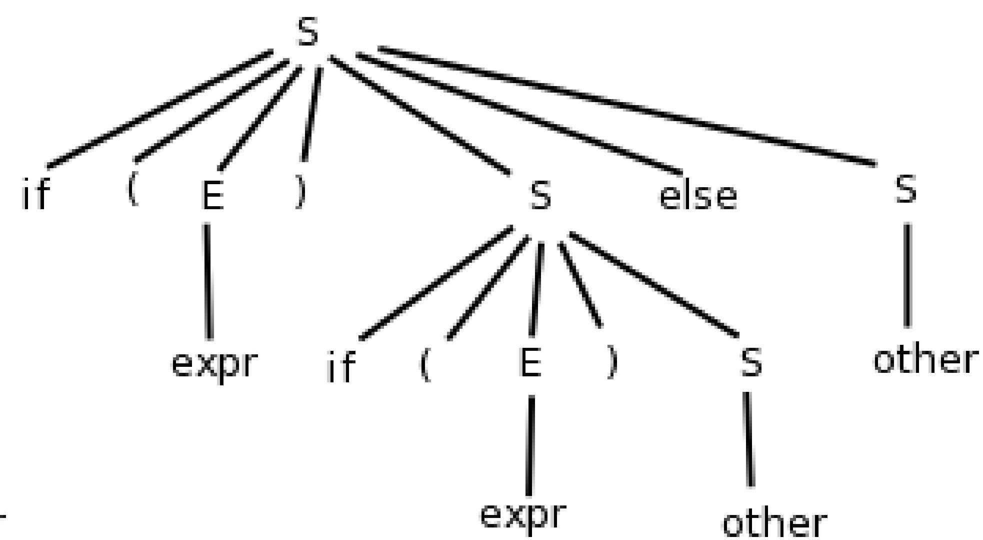
Parse Tree Example

- if (expr) if (expr) other else other

Production Rules

$S \rightarrow$ if (E) S / if (E) S else S
$S \rightarrow$ other
$E \rightarrow$ expr

Parse Tree 2



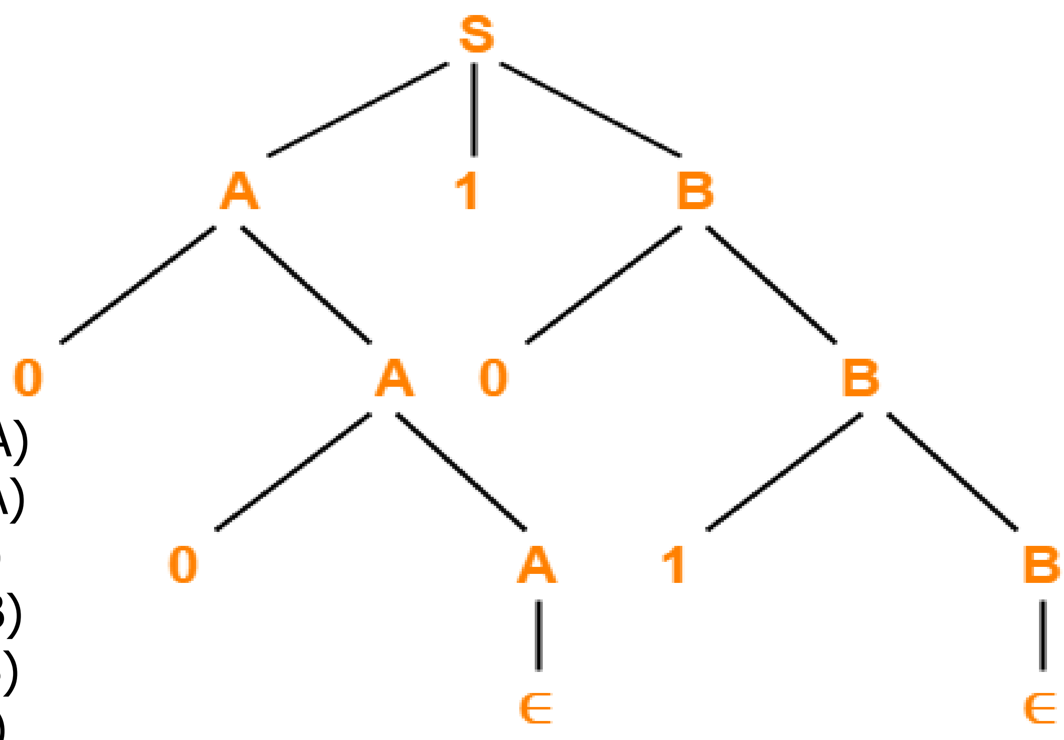
Derivation 2:

$S \Rightarrow$ if (E) S else S
$\Rightarrow$ if (E) S else other
$\Rightarrow$ if (E) if (E) S else other
$\Rightarrow$ if (E) if (E) other else other
$\Rightarrow$ if (E) if (expr) other else other
$\Rightarrow$ if (expr) if (expr) other else other

Parse Tree Example

- Write the leftmost derivation & rightmost derivation and draw the parse tree for the string w=00101 with the grammar {
  - $S \rightarrow A1B$
  - $A \rightarrow 0A \,/\, \epsilon$
  - $B \rightarrow 0B \,/\, 1B \,/\, \epsilon$}

$S \Rightarrow \mathbf{A}1B$	
$\Rightarrow 0\mathbf{A}1B$	(Using $A \rightarrow 0A$)
$\Rightarrow 00\mathbf{A}1B$	(Using $A \rightarrow 0A$)
$\Rightarrow 001\mathbf{B}$	(Using $A \rightarrow \epsilon$)
$\Rightarrow 0010\mathbf{B}$	(Using $B \rightarrow 0B$)
$\Rightarrow 00101\mathbf{B}$	(Using $B \rightarrow 1B$)
$\Rightarrow 00101$	(Using $B \rightarrow \epsilon$)



Summary

- Derivation using Grammar
- Types of derivation
- Examples
- Comparison of LM and RM derivation
- Definition of CFL
- Sentential Forms
- Parse Tree
- Examples for parse tree
- Yield of a parse tree

References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory*, Languages, and Computation, Pearson, 3rd Edition, 2011.
- Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class:

Ambiguity in Grammars & Languages

THANK YOU.