Modeling Computation

Introduction to Formal Languages and Automata

Parsing

Parsing

The first job of every compiler is to check syntactical correctness of a program.

CFGs provide compilers with the opportunity to parse programs in order to check for syntactical validity.

Example

$S \to I = E; S$

$S \to S\ \text{while}(C)\{S\}\ S$

$S \to \epsilon$

$I \to a \mid b \mid \cdots \mid z$

$C \to E > E$

$E \to (E) \mid -(E) \mid (E + E) \mid (E * E) \mid (E/E) \mid I$

$S \Rightarrow S\ \text{while}(C)\{S\}\ S$

$\Rightarrow I = E;\ S\ \text{while}(C)\{S\}\ S$

$\Rightarrow a = E;\ S\ \text{while}(C)\{S\}\ S$

$\vdots$

$\Rightarrow a = b;\ \text{while}(b > a)\{b = b - a$

Production rules can be applied in any order.

Thus, a string w can have many derivations.

Ex: derivation of $(a + (b * c))$

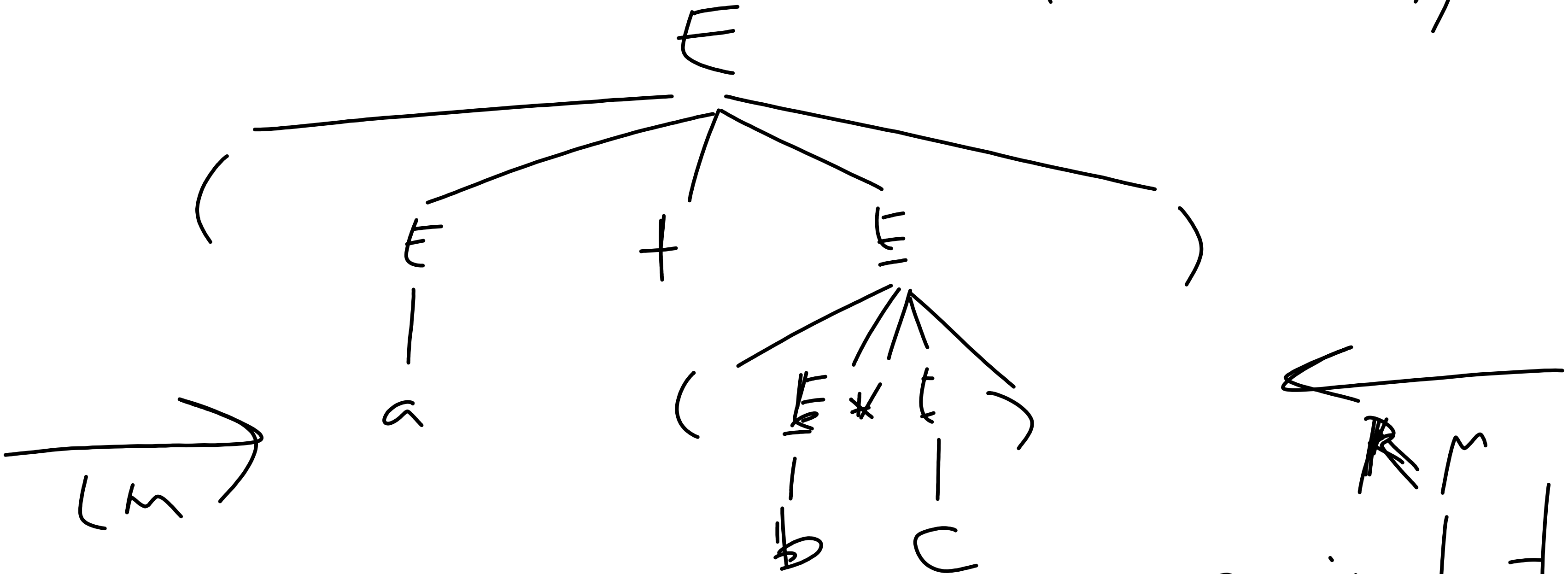
<u>Leftmost derivation:</u>

$E \Rightarrow (E + E) \Rightarrow (a + E) \Rightarrow (a + (E * E)) \Rightarrow (a + (b * c))$

<u>Rightmost derivation:</u>

$E \Rightarrow (E + E) \Rightarrow (E + (E * E)) \Rightarrow (E + (E * c)) \Rightarrow \cdot \cdot$

Visualize derivations with parse trees. (a+(b*c))



All derivations with the same parse tree are equivalent.

Different parse trees for the same word means that the grammar is ambiguous.

Example

$$S \to A \mid B \mid AB$$

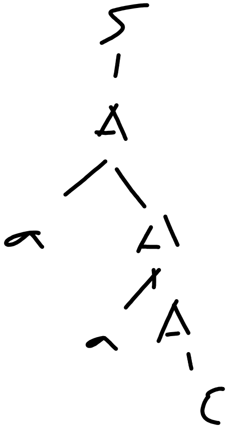
$$A \to aA \mid \epsilon$$

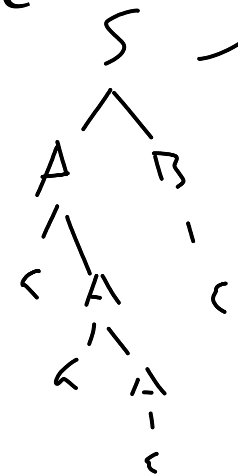
$$B \to bB \mid \epsilon$$

aa

$S \Rightarrow A \Rightarrow aA \Rightarrow aaA \Rightarrow aa$

$S \Rightarrow AB \Rightarrow aAB \Rightarrow aaAB \Rightarrow aaB \Rightarrow aa$





"Dangling Else" Example

<u>if</u> A <u>then</u> (<u>if</u> B (<u>then</u> C)) <u>else</u> D

1 2