CSE 396 Lecture Thu 3/12

"CSE 305:" Parse Trees, Ambiguity, Expression Grammars.

<u>Def</u>$^n$: A <u>parse</u> (sub-)<u>tree</u> for a CFG $G = (V, \Sigma, R, S)$ is a tree whose leaves are labeled by terminals and whose interior nodes are labeled by variables, such that for all interior nodes $v$ labeled $A$, if $X_1, \dots, X_k$ are the labels of the $k$ children of $v$, then (in left to right order)

$$A \to X_1 X_2 \cdots X_k \quad \text{is a rule in } R.$$

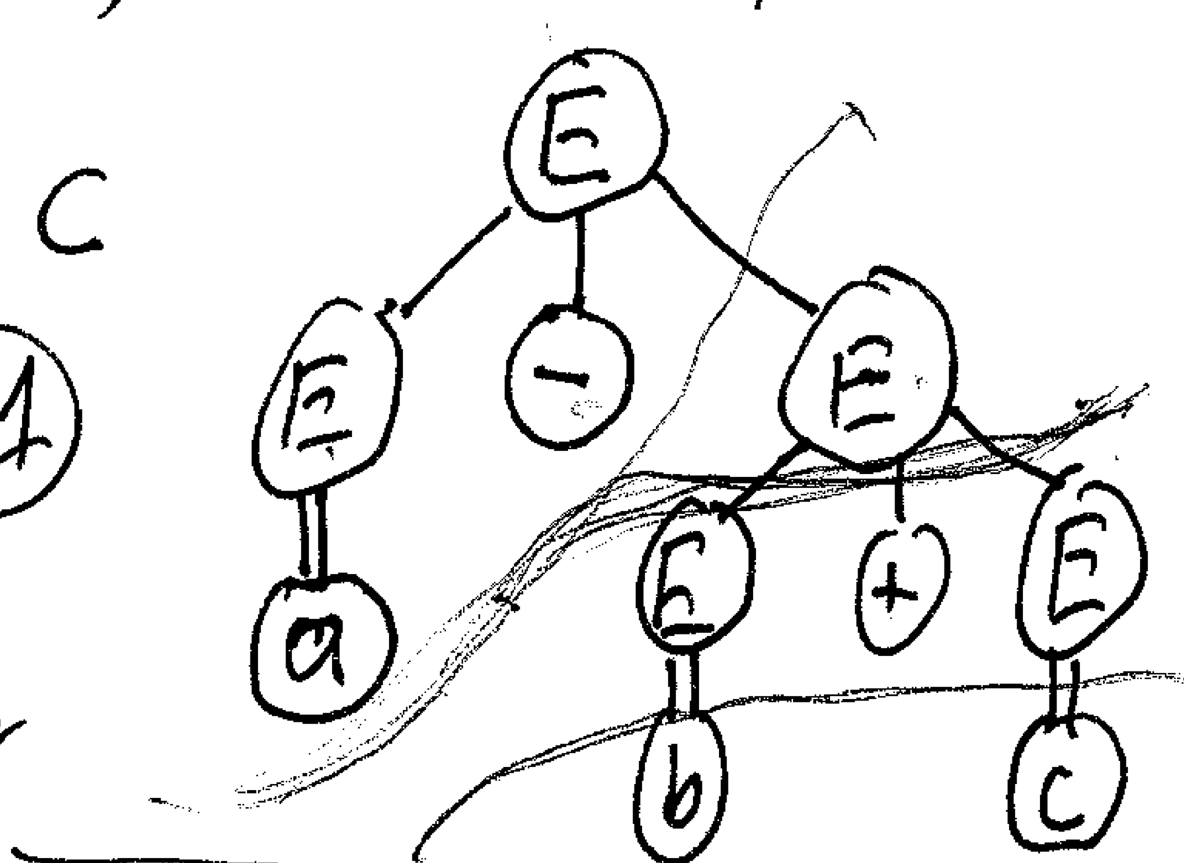
Parse tree has <u>S</u> at root, "subtree" has any variable at its root.
"Partial Parse Tree" allows some leaves to have variables (not yet expanded).

$$E \longrightarrow E+E \mid E-E \mid E*E \mid E/E \mid \underline{\text{const}} \mid \underline{\text{variable}} \mid (E)$$

$G = (V, \Sigma, R, S)$, $V = \{E\}$, $S = E$, $\Sigma = \{$ '(', ')', alphanumeric letters, digits, $\pm$ signs including $\}$ $\cup \{+, -, *, /\}$.

$a - b + c$

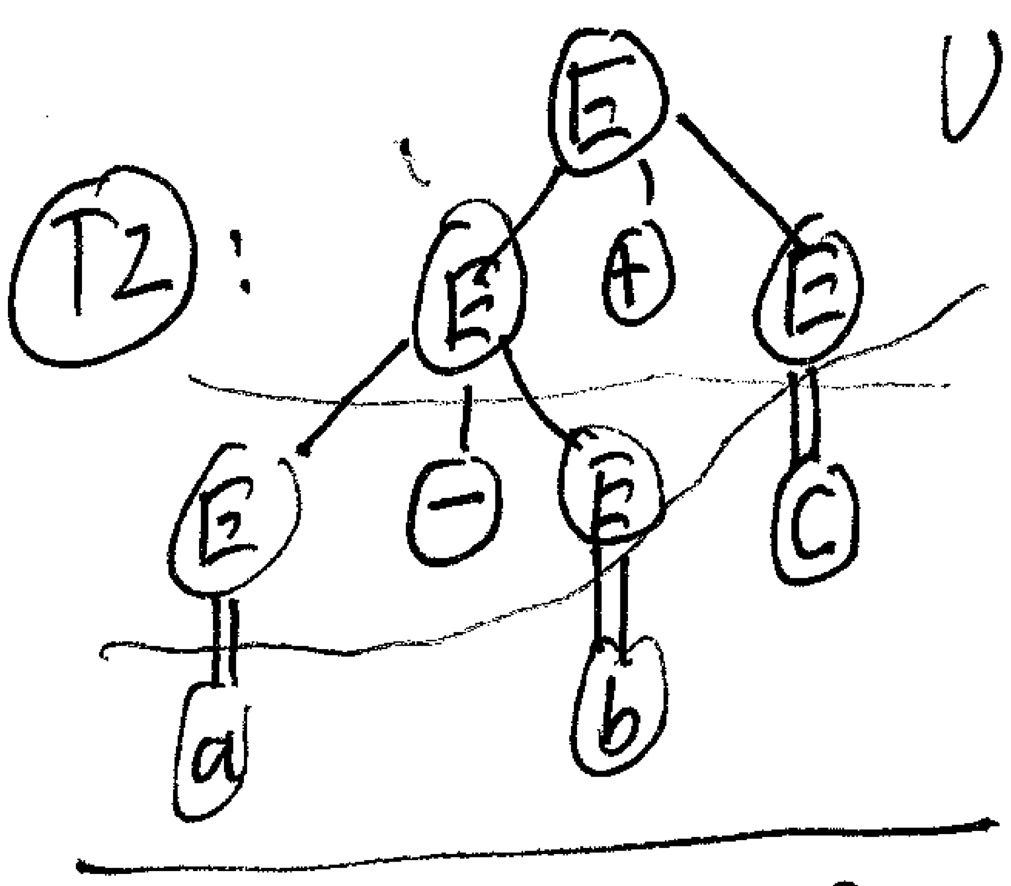
T1



$a \; - \; b \; + \; c$

Yield = leaves in L to R order

T2:



$a \; - \; b + c$

<u>Def</u>$^n$: A terminal string $x \in L(G)$ is <u>ambiguous</u> if $x$ has 2 different parse trees.

LM
T1 $E \Rightarrow E - E \Rightarrow a - E \Rightarrow a - E + E \Rightarrow a - b + E \Rightarrow a - b + c$

RM $E \Rightarrow E - E \Rightarrow E - E + E \Rightarrow E - E + c \Rightarrow E - b + c \Rightarrow a - b + c$

LM in T2 $E \Rightarrow E + E \Rightarrow E - E + E \Rightarrow a - E + E \Rightarrow a - b + E \Rightarrow a - b + c$

(*) To every parse tree there corresponds a unique <u>leftmost</u> derivation (And also a unique <u>RM</u> derivation)

Hence, equivalently, <u>$x$ is ambiguous in $L(G)$</u> if $x$ has two different leftmost derivations.

$\therefore$ $a - b + c$ is ambiguous

$a/b \cdot c \stackrel{?}{=} \frac{a}{bc}$

We *can* derive unambiguous readings in $G$.

*Issue* is that this doesn't force unambiguous strings.

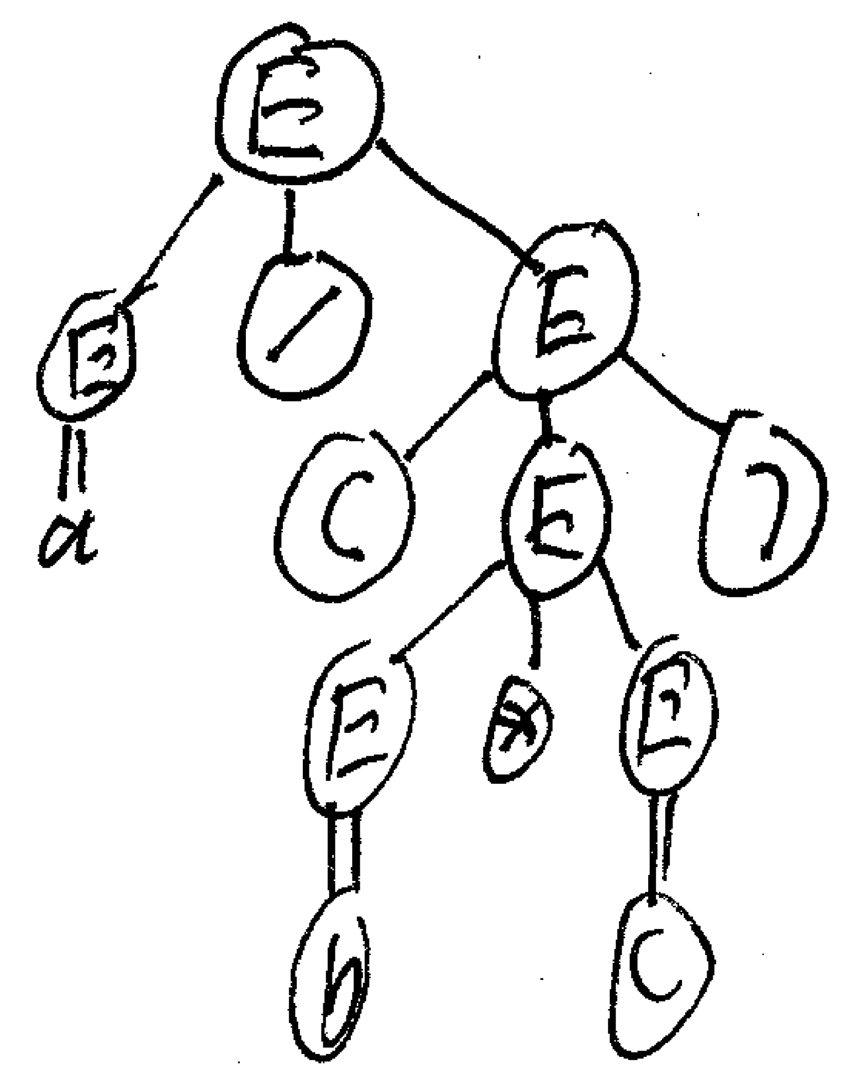
Expression grammars can force it by forcing parentheses:

$$E \Rightarrow (E+E) \mid (E-E) \mid (E*E) \mid (E/E) \mid const \mid var \mid$$

$$E \Rightarrow (E/E) \Rightarrow (a/E) \Rightarrow (a/(E*E))$$

$$\Rightarrow (a/(b*E)) \Rightarrow (a/(b*C))$$

problem: this is not the same string.



$a / ( b * C )$

We *can* force unambiguous parsing of all humanly natural expressions, if we use more variables. "A Term-Factor Grammar With Cascading Precedence."

$V = \{E, T, F\}$ $\quad E \Rightarrow E+T \mid E-T \mid T$

$S = E$ $\quad T \Rightarrow T*F \mid T/F \mid F$

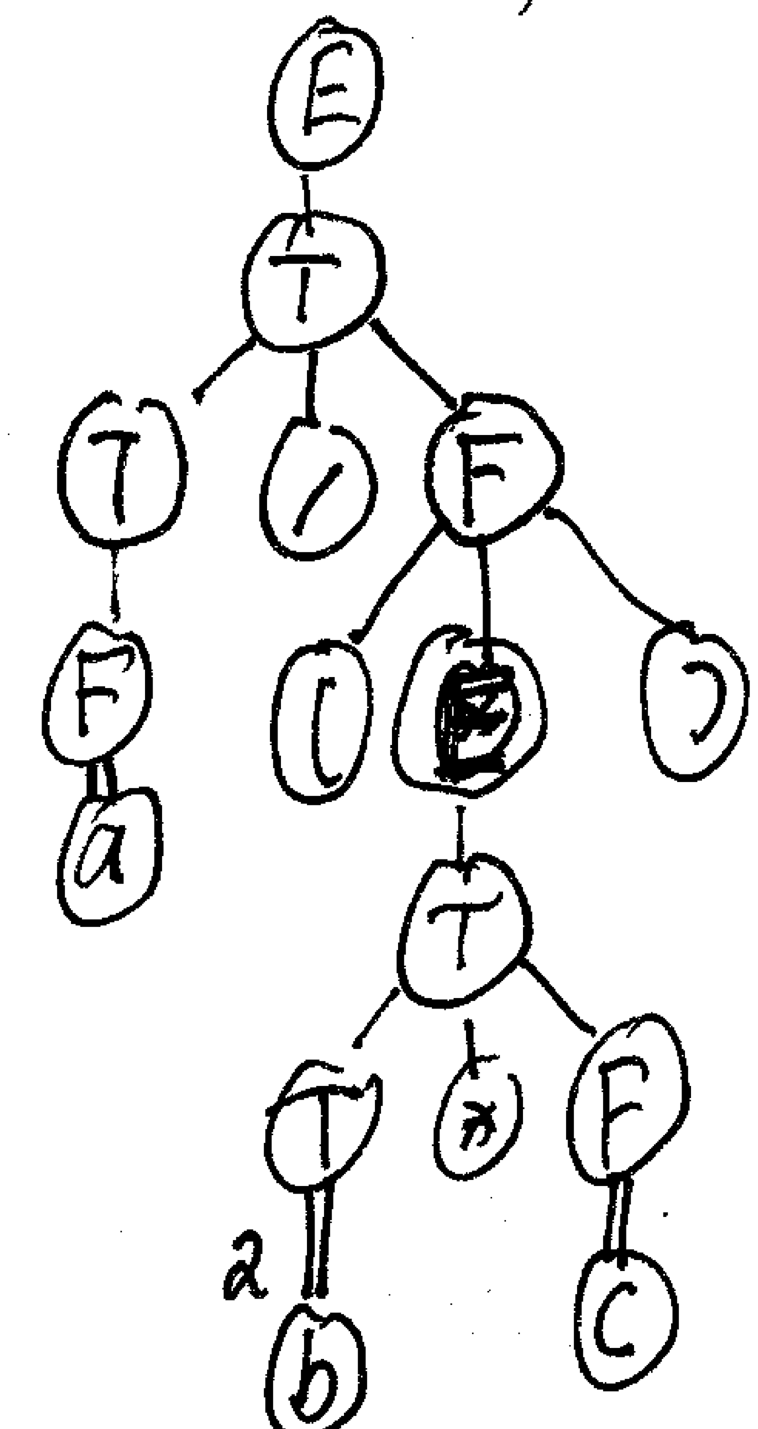
$\Sigma$ same as before. $\quad F \rightarrow$ *const* $\mid$ *variable* $\mid (E)$

The corresponding LM derivation, with abbreviated steps.

$$E \Rightarrow T \Rightarrow T/F \Rightarrow^2 a/F$$

$$\Rightarrow a/(E) \Rightarrow a/(T) \Rightarrow a/(T*F)$$

$$\Rightarrow^* a/(b*F) \Rightarrow a/(b*C).$$



$a / ( b * C )$

*Fact* (not proved in text) This $G$ is *unambiguous*, meaning *every* $x \in L(G)$ has a unique parse tree.

$a / b * C$:

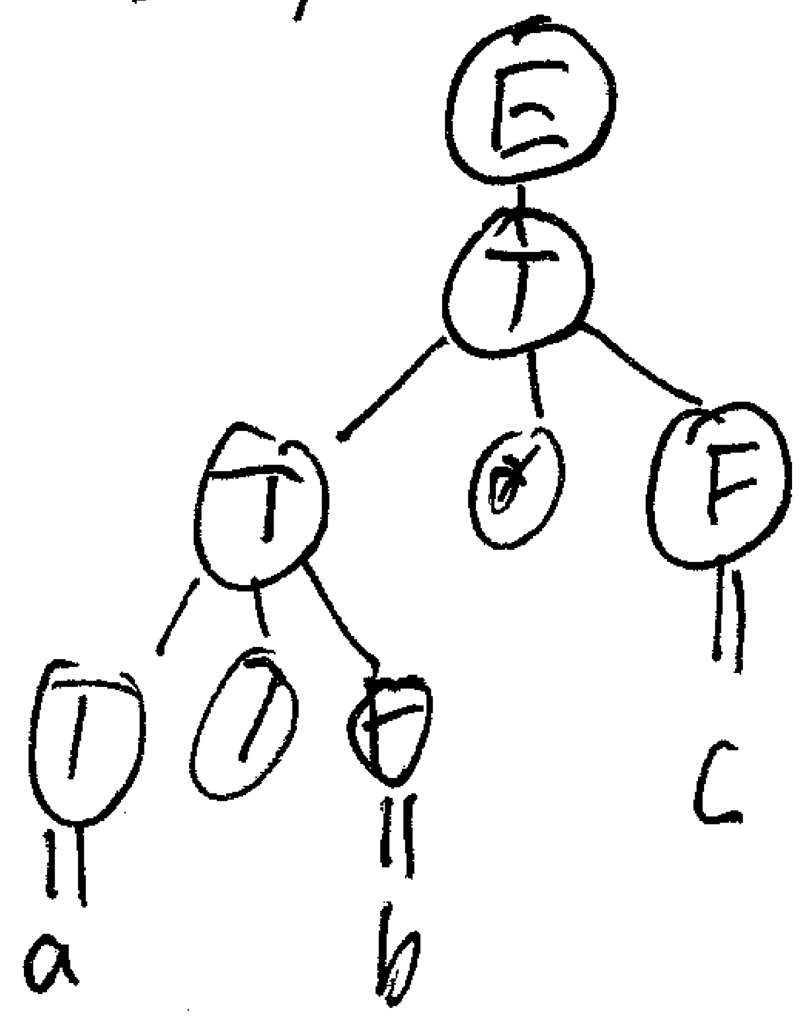
I could replace $E \rightarrow E+T$ by

$E \rightarrow EX$

$X \rightarrow PT$

$P \rightarrow +$

legal in Chomsky Normal Form.



$\underbrace{a \;/\; b}_{\text{grouped}} * C$ by the rule of *left associativity*. Hence $\dfrac{a}{b} * C$

Note: $E \rightarrow T$ and $T \rightarrow F$ are *unit productions*.

We could *extend* the grammar via rules for IDs.

$F \rightarrow \langle letter \rangle \mid \langle letter \rangle \cdot A$ $\quad A$ = means alphanum

$A \rightarrow \langle letter \rangle A \mid \langle digit \rangle A \mid \_A \mid \langle letter \rangle \langle digit \rangle \mid \_$

$\langle letter \rangle \rightarrow a \mid b \mid \cdots \mid z \mid A \mid \cdots$ etc. $\quad \langle digit \rangle \rightarrow 0 \mid 1 \mid \cdots \mid 9.$