Def: A parse tree 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 $X_1, \ldots, X_k$ are the labels of the $k$ children of $v$, then $A \rightarrow X_1X_2\ldots X_k$ is a rule in $R$.

A parse tree has $S$ at root, "subtree" has any variable at its root. "Partial Parse Tree" allows some leaves to have variables (not yet expanded).

$$E \rightarrow E + E \mid E - E \mid E \ast E \mid E / E \mid \text{const} \mid \text{variable} \mid (E)$$

$$G = (V, \Sigma, R, S), \ V = \{E, T\}, \ S = E, \ \Sigma = \{\ast, \pm, \), \}, \ \text{letters, } (d|9|8), \ \pm \text{ signs}\}$$

$$a - b + c$$

Yield = leaves in L to R order

```
E \rightarrow E - E \Rightarrow a - E \Rightarrow a - E + E \Rightarrow a - b + c
```

```
E \rightarrow E - E \Rightarrow E - E + E \Rightarrow E - b + c \Rightarrow a - b + c
```

```
E \rightarrow E + E \Rightarrow E - E + E \Rightarrow a - E + E \Rightarrow a - b + E \Rightarrow a - b + c
```

Def: A terminal string $x \in L(G)$ is ambiguous if $x$ has two different parse trees.

```
\begin{array}{c}
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\end{array}
```

To every parse tree there corresponds a unique leftmost derivation.

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

```
\begin{array}{c}
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\end{array}
```

And also a unique rightmost derivation.

```
\begin{array}{c}
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\end{array}
```

$a - b + c$ is ambiguous, leftmost derivations.

```
\begin{array}{c}
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\text{LM} \downarrow \\
\end{array}
```

$a - b + c$ is ambiguous, leftmost derivations.
We can derive unambiguous readings in $G$. Issue is that this doesn't force unambiguous strings. Expression grammars can force it by forcing parentheses:

$$E \rightarrow (E+E) \mid (E-E) \mid (E \times E) \mid (E/ E) \mid \text{const} \mid \text{var}$$

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

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

Problem: this is not the same string.

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

$$V = \text{E, T, F, b, c}$$

$$E \rightarrow E+T \mid E-T \mid T$$

$$T \rightarrow T*F \mid T/F \mid F$$

$$\Sigma$$ same as before.

$$F \rightarrow \text{const} \mid \text{variable} \mid (E)$$

The corresponding LM derivation, with abbreviated steps:

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

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

$$\Rightarrow a/(b*F) \Rightarrow a/(b*c)$$

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

$$a/(b\ast c)$$

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

We could extend the grammar via rules for PEs:

$$F \rightarrow \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \cdot A \mid A \Rightarrow \langle \text{letter} \rangle \cdot A \mid \langle \text{digit} \rangle \cdot \cdot \cdot$$

$$\langle \text{digit} \rangle \rightarrow \text{a|b|c|1|2|3|4|5|6|7|8|9}$$

legal in

Chemistry Normal Form.

$$\frac{a}{b \ast c}$$

by the rule of left associativity. Hence $\frac{a}{b \ast c}$