Parens trees

Given some fixed grammar $G$, a parse tree of $G$ is a rooted, ordered tree whose nodes are labeled with grammar symbols, such that:
- Every internal node $n$, $n$'s label is a nonterminal:
  - $\text{ACN}$
  - and the children of $n$ are
  - form the body of some production whose head is $A$
  - in $G$, $A \rightarrow X_1 \ldots X_n$ is a prod.

A exception: if $A \rightarrow e$ is a production, then using this would give:

- $e$

All leaves are labeled with terminals or $e$.

The yield of a parse tree is the string of terminals obtained by concatenating all the leaves from left to right:

$\text{[accumulate E "disappear"]}$

$E \rightarrow \{ E \} | E + E | E - E | (E)$

Parse tree for $E + E - E$

Todo:
- For any string $w$, if it is derivable from some symbol in $G$, $w$ and only if $w$ is a complete parse tree yielding $w$.
- Further, complete parse trees and complete leftmost derivations are in one-to-one correspondence.

Claim:
- A grammar $G$ is ambiguous if:
  - there is some string $w$ that can be derived in two different complete parse trees.

For help:

Job of a parser:
- To "comprehend" a parse tree for the input.
- If there is no $\text{[look for E]}$

$L(G) = \{ w \mid w \text{ is a string derivable from } S \}$

$e \in (G)$
Syntax-directed translation:
produce output translation as input is parsed

2 types of annotating a grammar:
- syntax-directed definition (SDD)
- syntax-directed translation scheme (SDTS)

\[ c \times (c + c) \]

\[ (3 \times 17) + 9 \]

\[ E \rightarrow T + E \rightarrow F * T \rightarrow T * F \rightarrow (c) \]

\[ E \rightarrow (c) \]

\[ F \rightarrow (c) \]

\[ T \rightarrow (c) \]

\[ c \times c \]