Context-free Grammars (Languages)

Key Definition (CFG): A context-free grammar is a 4-tuple (V, Σ, R, S) such that:

1. V is a finite set of **variables**,
2. Σ is a finite set of **terminals** (disjoint with V),
3. R is a finite set of **rules** of the form $R_i \to \alpha$ (a single variable on the left, and an expression on the right), and
4. $S \in V$ is the start variable.

Key Definition (CFL): A language, L is context-free if some CFG produces it. That is, there must exist a CFG, G, such that $L = L(G)$.

Keywords: derivation, production, rule, parse tree

<u>Example</u>: As we saw before, $\{0^n 1^n | n \geq 0\}$ is not a regular language, but it is context-free. Here is a CFG for it:

$$S \to 0S1|\epsilon.$$

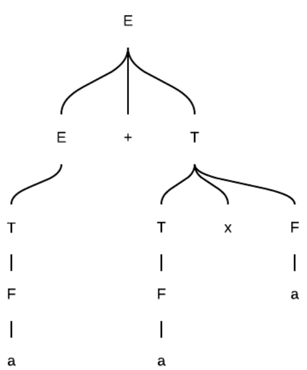
<u>Example</u>: The language of valid arithmetic expressions can be expressed as the following CFG, where a is any number, E denotes an expression, F denotes a factor, and T denotes a term:

$$E \to E + T|T$$
$$T \to T \times F|F$$
$$F \to (E)|a$$

There is a **derivation** for expression $a + a \times a$ that can be obtain via a series of **productions** like so:

$$E \Rightarrow E + T \Rightarrow T + T \Rightarrow F + T \Rightarrow a + T \Rightarrow a + T \times F \Rightarrow a + F \times F \Rightarrow a + a \times F \Rightarrow a + a \times a$$

The **parse tree** for the dervivation would be (and you should convince yourself that only one parse tree is possible):



<u>Example</u>: (ambiguity) The following language also generates valid arithmetic expressions, however, unlike the previous grammar, this one is **ambiguous**. Ambiguous means there are two (or more) distinct parse trees for some strings, or two (or more) **left-most derivations**. A left-most derivation is one in which we only replace left-most variables during a production.

$$E \to E + E|E \times E|(E)|a$$

<u>**Try it!**</u> Give me the parse trees for "a + a x a" using each of the grammars. Note how there is more than one parse tree using the latter grammar – it is an **ambiguous** grammar.