Parsing
The derivation of a string produces a parse tree for the string:

Grammar:

\[
\begin{align*}
E & \rightarrow E+T \mid E-T \mid T \\
T & \rightarrow T*F \mid T/F \mid F \\
F & \rightarrow (E) \mid G \\
G & \rightarrow G \text{ digit} \mid \text{ digit}
\end{align*}
\]

Derivation:

\[
\begin{align*}
E & \rightarrow T \\
& \rightarrow T*F \\
& \rightarrow F*F \\
& \rightarrow G*F \\
& \rightarrow 3*F \\
& \rightarrow 3*(E) \\
& \rightarrow 3*(E+T) \\
& \rightarrow 3*(T+T) \\
& \rightarrow 3*(F+T) \\
& \rightarrow 3*(G+T) \\
& \rightarrow 3*(4+T) \\
& \rightarrow 3*(4+F) \\
& \rightarrow 3*(4+G) \\
& \rightarrow 3*(4+5)
\end{align*}
\]

Parse Tree:
Example 1: Find a grammar for \( \{0^n1^n \mid n \geq 0 \} \) This is one of the languages we showed isn't regular.

\[
S \rightarrow 0 \ S \ 1 \mid \varepsilon
\]

Example 2: Find a grammar for \( \{ww^{\text{rev}} \mid w \in (0+1)^* \} \) (even-length palindromes)

\[
S \rightarrow 0 \ S \ 0 \mid 1 \ S \ 1 \mid \varepsilon
\]

Example 3: Find a grammar for the language of all palindromes of 0's and 1's

\[
S \rightarrow 0 \ S \ 0 \mid 1 \ S \ 1 \mid 0 \mid 1\mid \varepsilon
\]
Note that we can reproduce the string being parsed with a left-to-right traversal of the leaves of the parse tree:

\[ 3^* (4+5) \]
Regular Grammars

Consider the DFA

Here is a grammar for the language this accepts:

\[
S \rightarrow 1T \mid 0U \\
T \rightarrow 0T \mid 1U \\
U \rightarrow 0S \mid \varepsilon
\]
Here is a derivation of 00101:

1. \( S \Rightarrow 0U \)
2. \( \Rightarrow 00S \)
3. \( \Rightarrow 001T \)
4. \( \Rightarrow 0010\overline{T} \)
5. \( \Rightarrow 00101U \)
6. \( \Rightarrow 00101 \)
Definition: A grammar that has only rules of the forms
- $X \Rightarrow a Y$
- $X \Rightarrow \varepsilon$

is called a *regular grammar*. 
**Theorem:** The language defined by a regular grammar is regular. All regular languages are defined by regular grammars.

**Proof:** Given a regular grammar, build an NFA with the non-terminal symbols as states and transition function $\delta$ defined by: if $X \Rightarrow a Y$ is a grammar rule then $\delta(X,a)$ includes $Y$. If there is a rule $X \Rightarrow \varepsilon$ then $X$ is a final state in the NFA. Note that every step of derivation with the grammar has the form $S \Rightarrow \alpha Y$ where $\alpha$ is a string containing only terminal symbols. An easy induction on the length of $\alpha$ shows that $S \Rightarrow \alpha Y$ if and only if the string $\alpha$ takes the automaton from its start state to state $Y$. This means the automaton accepts the same strings as are generated by the grammar.
On the other hand, if we start with a regular language it must have a DFA that accepts it. We can generate a regular grammar from this DFA. Again, a straightforward induction shows that the grammar defines the same language as the automaton.

Since regular grammars are context-free, we see that all regular languages are context free. But the family of context-free languages includes many languages that are not regular, including

$$\{0^n1^n \mid n \geq 0\} \quad \{ww^{rev} \mid w \in (0+1)^*\}$$