Grammars II

See Section 5.2

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

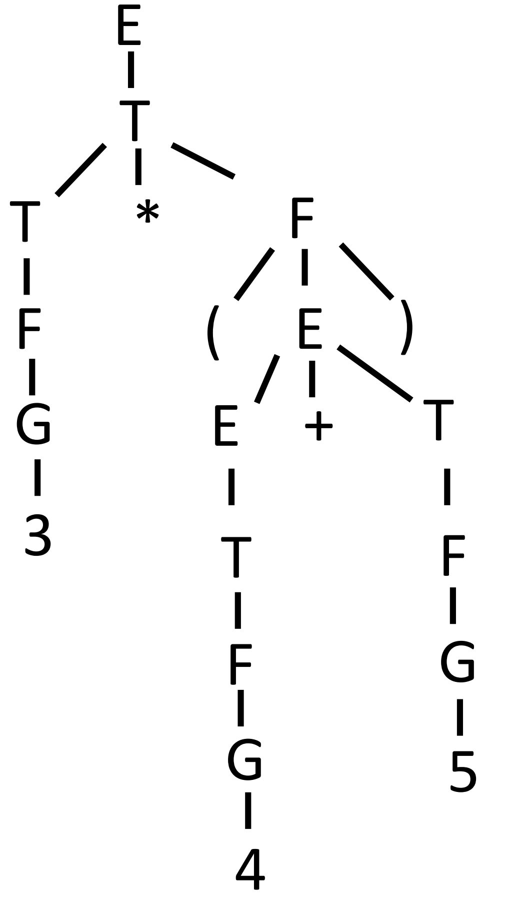
Grammar:

E => E+T | E-T | T
T => T*F | T/F | F
F => (E) | G
G => G digit | digit

Derivation:

E => T
=> T*F
=> F*F
=> G*F
=> 3*F
=> 3*(E)
=> 3*(E+T)
=> 3*(T+T)
=> 3*(F+T)
=> 3*(G+T)
=> 3*(4+T)
=> 3*(4+F)
=> 3*(4+G)
=> 3*(4+5)

Parse Tree:



Example 1: Find a grammar for $\{0^n1^n \mid n>= 0\}$ This is one of the languages we showed isn't regular.

$S => 0 S 1 \mid \varepsilon$

Example 2: Fina a grammar for $\{0^n2^m1^n \mid n, m >= 0\}$

$s => 0 S 1 \mid T$

$T => 2 T \mid \varepsilon$

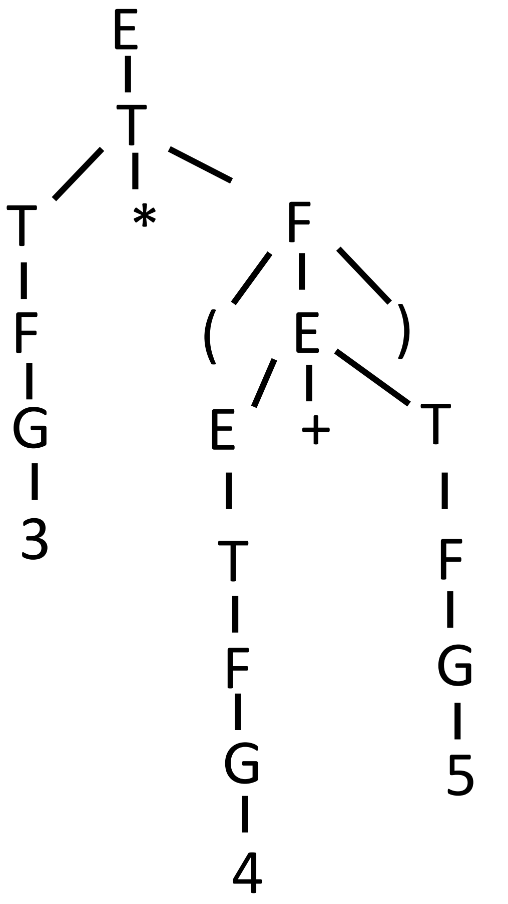
Example 3: Find a grammar for $\{ww^{rev} \mid w \in (0+1)^* \}$ (even-length palindromes)

$S => 0 S 0 \mid 1 S 1 \mid \varepsilon$

Example 4: Find a grammar for the language of all palindromes of 0's and 1's

$S => 0 S 0 \mid 1 S 1 \mid 0 \mid 1 \mid \varepsilon$

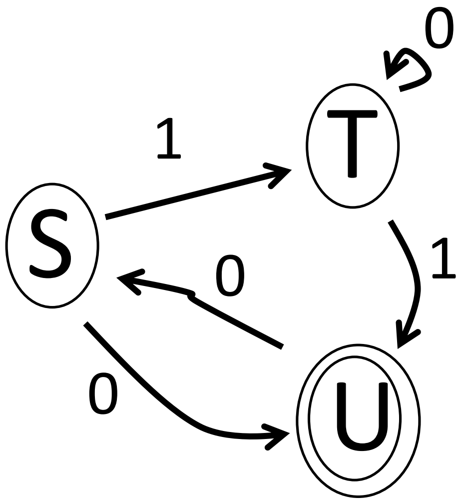
Note that we can reproduce the string being parsed with a left-to-right traversal of the leaves of the parse tree:



3*(4+5)

Regular Grammars

Consider the DFA

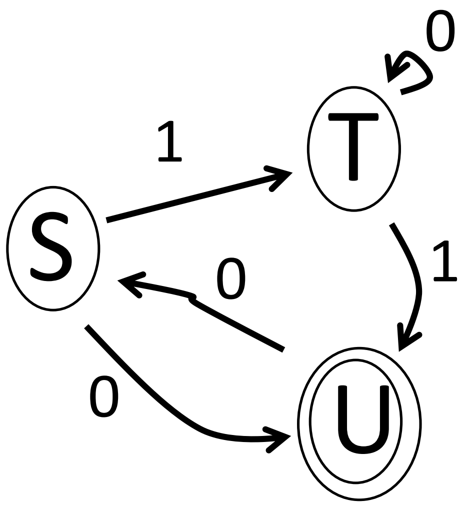


Here is a grammar for the language this accepts:

S => 1T | 0U

T => 0T | 1U

U => 0S | ε



S => 1T | 0U
T => 0T | 1U
U => 0S | ε

Here is a derivation of 00101:

S => 0<u>U</u>
=> 00<u>S</u>
=> 001<u>T</u>
=> 0010<u>T</u>
=> 00101<u>U</u>
=> 00101

Definition: A grammar that has only rules of the forms

- X => a Y
- X => a

is called a *regular grammar*.

For example, here is a regular grammar:

S => 0S | 1T | 0
T => 0T | 1S | 1

A typical derivation is S => 0S=>00S=>001T=>0010T=>00101

Theorem: The language defined by a regular grammar is regular.

Proof: Given a regular grammar, build an NFA from it. The states of the NFA are the non-terminal symbols of the grammar, plus an extra final state called "Accept". If X=>aY is a rule in the grammar add a transition in the NFA δ(X,a) =Y. If X=>a is a grammar rule make a transition δ(X,a)=Accept.

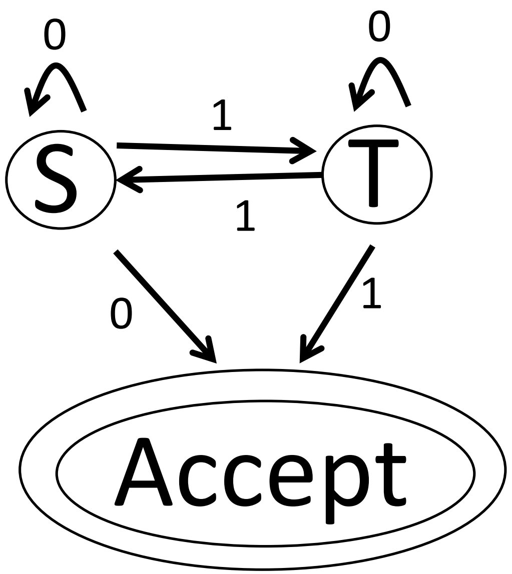
Every step except the last of a derivation of a string in the regular grammar has the form $S \stackrel{*}{\Rightarrow} \alpha X$. An easy induction shows that $S \stackrel{*}{\Rightarrow} \alpha X$ if and only if string α takes the NFA from state S to state X. The grammar derives string w if and only if w = αa and there is a non-terminal symbol X with $S \stackrel{*}{\Rightarrow} \alpha X$ and X => a, which says the string αa will take the NFA to state Accept. This says the grammar derives string w if and only if the NFA accepts the string w.

For example, start with the regular grammar

S => 0S | 1T | 0

T => 0T | 1S | 1

This gives the NFA



Both the grammar and the NFA describe strings with an even number of 1's.

Theorem: Every regular language has a regular grammar.

Proof: Start with DFA that describes a regular language. We will build a grammar for the language. The non-terminal symbols of the grammar will be the names of the states of the DFA. If the DFA has transition δ(X,a) =Y, add the grammar rule X => aY. If the DFA has transition δ(X,a) =Y and Y is a final state, also add the grammar rule X => a. A string w = αa is accepted by the DFA if and only if S $\overset{*}{\Rightarrow}$ αX and X=>a, so w is accepted by the DFA if and only if S $\overset{*}{\Rightarrow}$ w.

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

$\{0^n1^n \mid n >= 0\}$ and $\{ww^{rev} \mid w \in (0+1)^*\}$