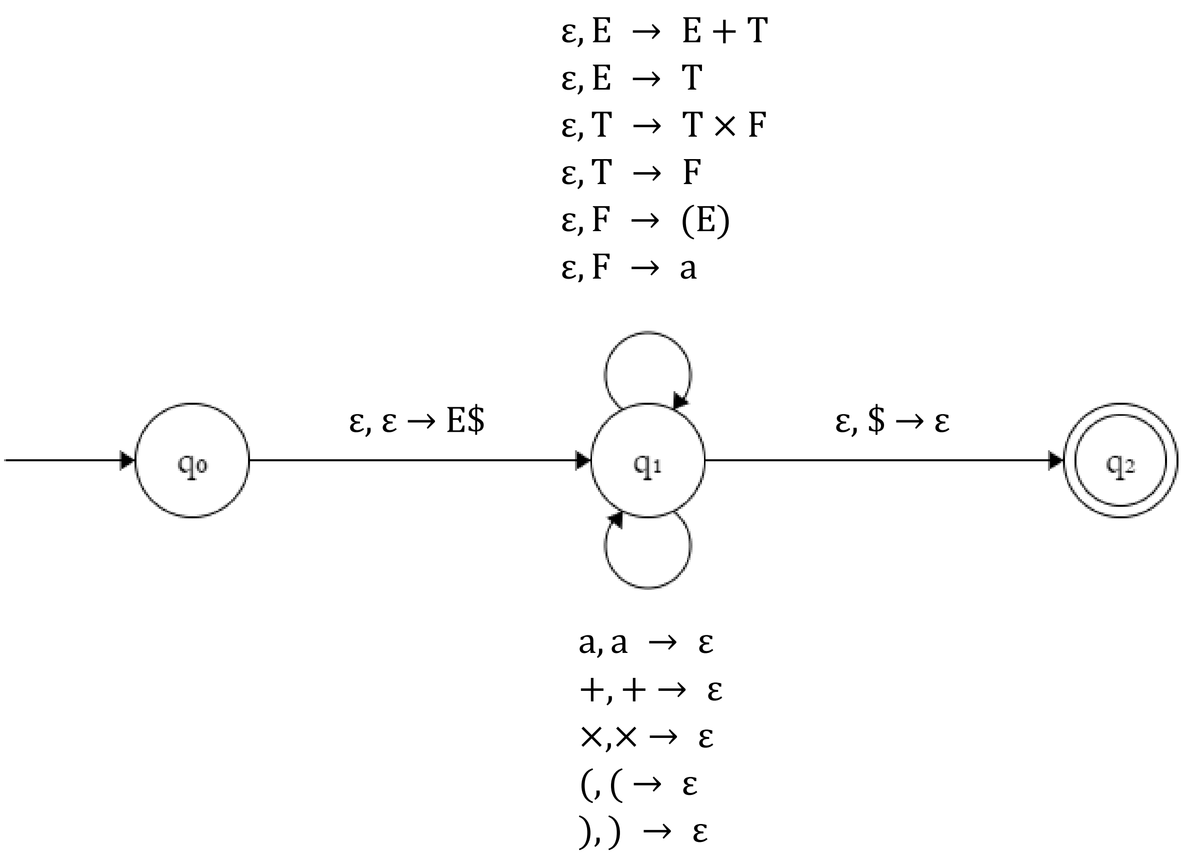
CMPS 130 Homework 8

Textbook Exercises

2.11. $E \to E + T \mid T$
$T \to T \times F \mid F$
$F \to (E) \mid a$

Think of the upper transition arrows in the middle state as following the production rules to generate any string in the CFL, and place it on the stack. The lower transition arrows in the middle state are simply processing the string (from left to right) and only moving to the next letter if the current one is on the stack. If a string is in this CFL, it is possible for it to be generated by the production rules and placed on the stack. Eventually these strings will be read all the way through and reach the bottom of the stack ($ symbol), where they enter the accept state.



2.14. Original CFG:

$A \to BAB \mid B \mid \varepsilon$
$B \to 00 \mid \varepsilon$

Add rules for ε-productions ($A \to \varepsilon$, $B \to \varepsilon$):

$A \to BAB \mid AB \mid BA \mid BB \mid B \mid \varepsilon$
$B \to 00 \mid \varepsilon$

Add rules for unit productions ($A \to B$):

$A \to BAB \mid AB \mid BA \mid BB \mid B \mid 00 \mid \varepsilon$
$B \to 00 \mid \varepsilon$

Remove ε and unit productions ($A \to B \mid \varepsilon$, $B \to \varepsilon$):

$A \to BAB \mid AB \mid BA \mid BB \mid 00$
$B \to 00$

Add variables for terminals in strings of two or more symbols:

$A \to BAB \mid AB \mid BA \mid BB \mid V_0V_0$
$B \to V_0V_0$
$V_0 \to 0$

Reduce remaining productions to no more than two variables:

$A \to BC \mid AB \mid BA \mid BB \mid V_0V_0$
$B \to V_0V_0$
$V_0 \to 0$
$C \to AB$

Textbook Problems

2.30 a. $L = \{0^n 1^n 0^n 1^n \mid n \geq 0\}$

Demon picks p

I pick $s \in L = 0^p 1^p 0^p 1^p$ s.t. $|s| \geq p$

Demon picks $uvwxy = s$ s.t. $|vwx| \leq p$ and $vx \neq \varepsilon$

I pick $i = 2$

Because $|vwx| \leq p$, vwx is either entirely in a 0^p group or a 1^p group, or straddles a 0^p group and a 1^p group. In either case, you are only pumping one 0^p group and/or one 1^p group (and never pumping both 0^p groups or both 1^p groups). Therefore, the resulting string uv^2wx^2y will always contain one $0^{>p}$ group and one 0^p group, and/or one $1^{>p}$ group and one 1^p group, which is clearly not in the language.
$\therefore$ L is not context free.

d. $L = \{t_1\#t_2\# \ldots \#t_k \mid k \geq 2, \text{each } t_i \in \{a, b\}^*, \text{ and } t_i = t_j \text{for some } i \neq j\}$

Demon picks p

I pick $s \in L = a^p b^p \# a^p b^p$ s.t. $|s| \geq p$

Demon picks $uvwxy = s$ s.t. $|vwx| \leq p$ and $vx \neq \varepsilon$

I pick $i = 2$

Because $|vwx| \leq p$, vwx is either entirely in an $a^p b^p$ group or is somewhere in the substring $b^p \# a^p$. In the first case, pumping one $a^p b^p$ group but not the other results in a string that is not in L. In the second case, pumping any letter in the substring $b^p \# a^p$ results in a string that is not in L.
$\therefore$ L is not context free.

2.31. Let B be the language of all palindromes over {0,1} containing equal numbers of 0s and 1s.

Demon picks p

I pick $s \in B = 0^p 1^{2p} 0^p$ s.t. $|s| \geq p$

Demon picks $uvwxy = s$ s.t. $|vwx| \leq p$ and $vx \neq \varepsilon$

I pick $i = 2$

Because $|vwx| \leq p$, vwx is either entirely in a 0^p group or entirely in 1^{2p}, or straddles between a 0^p group and the 1^{2p} group. In the first case, pumping results in a string with an unequal number of 0s and 1s that is not in the language. In the second case, pumping results in a string that is clearly not a palindrome because there are more 0s on one side than the other, or you will once again end up with a string with an unequal number of 0s and 1s. Both of these are not in the language.

∴ B is not context free.

2.35. Let G be a CFG in CNF with b variables. If G generates some string with a derivation having at least 2^b steps, L(G) is infinite.

Proof: Since G is in CNF, its derivation tree is a binary tree (with terminals as leaf nodes, and variables as non-leaf nodes). If a string in G requires exactly 2^b steps, the derivation tree contains 2^b+1 nodes (the +1 comes from the start variable). Therefore, any string that requires at least 2^b steps has greater or equal than 2^b+1 nodes. The height of this derivation tree is at least b+1 (height of a full binary tree is $\lceil \log_2(n+1) \rceil$). This means that there were at least b+1 productions. By the pigeonhole principle, a variable was seen at least twice in this path (since there are only b variables). Going back to the derivation tree, you can replace the "lower" instance of this variable with the "higher" instance of this variable in the tree to produce another string. You can do this an infinite number of times. Therefore, L(G) is infinite.