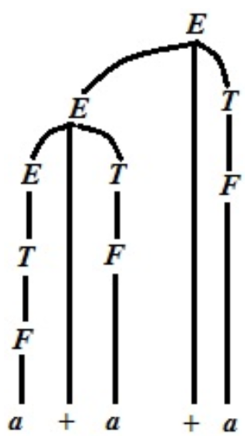
CS420 HW2 Solutions

2.1)

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

Give parse trees and derivations for each string.

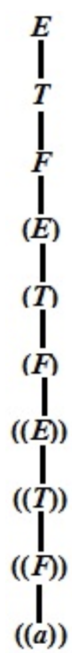
2.1.c) $a + a + a$



The derivation is as follows

$$
\begin{aligned}
E &\implies E+T \\
&\implies E+T+T \\
&\implies T+T+T \\
&\implies F+T+T \\
&\implies F+F+T \\
&\implies F+F+F \\
&\implies a+F+F \\
&\implies a+a+F \\
&\implies a+a+a
\end{aligned}
$$

2.1.d) $((a))$



$E \implies T \implies F \implies (E) \implies (T) \implies (F) \implies ((E)) \implies ((T)) \implies ((F)) \implies ((a))$

2.4.e) Give context-free grammars that generate the following languages.The alphabet $\sum$ is $\{0,1\}$. $\left\{w|w = w^R, \text{ that is, } w \text{ is a palindrome}\right\}$

The context free gramar for the language is given by $S \to 0|1|0S0|1S1|\epsilon$

2.6.b) Give context-free grammars generating the following language, The complement of the language $\{a^n b^n | n \geq 0\}$.

The compliment of the language L either should start with b or end in a, or if it starts with a and ends with b the substring between the two must not be in L. So we can write the grammar as follows.

$$S \to bA|Aa|aSb$$
$$A \to aA|bA|\epsilon$$

2.9) Give a context-free grammar that generates the language

$$A = \left\{a^i b^j c^k | i = j \text{ or } j = k \text{ where } i, j, k \geq 0\right\}$$

We can split A into two languages $A_1 = \left\{a^i b^j c^k | i = j \text{ where } i, j, k \geq 0\right\}$ and $A_2 = \left\{a^i b^j c^k | j = k \text{ where } i, j, k \geq 0\right\}$

CFG for the language A_1 is as follows

$$S_1 \to S_1 c|E|\epsilon$$
$$E \to aEb|\epsilon$$

CFG for the language A_2 is as follows

$$S_2 \to aS_2|F|\epsilon$$
$$F \to bFc|\epsilon$$

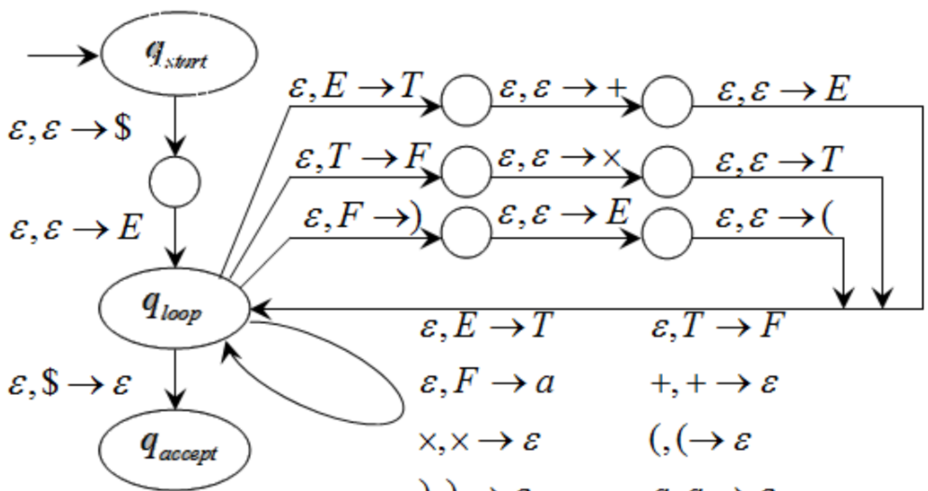
We can take the union of the two CFG's for the language A

$$S \to S_1|S_2$$

For generating the string $w = a^n b^n c^n$ we can use either S_1 or S_2, therefore the grammar is ambigous.

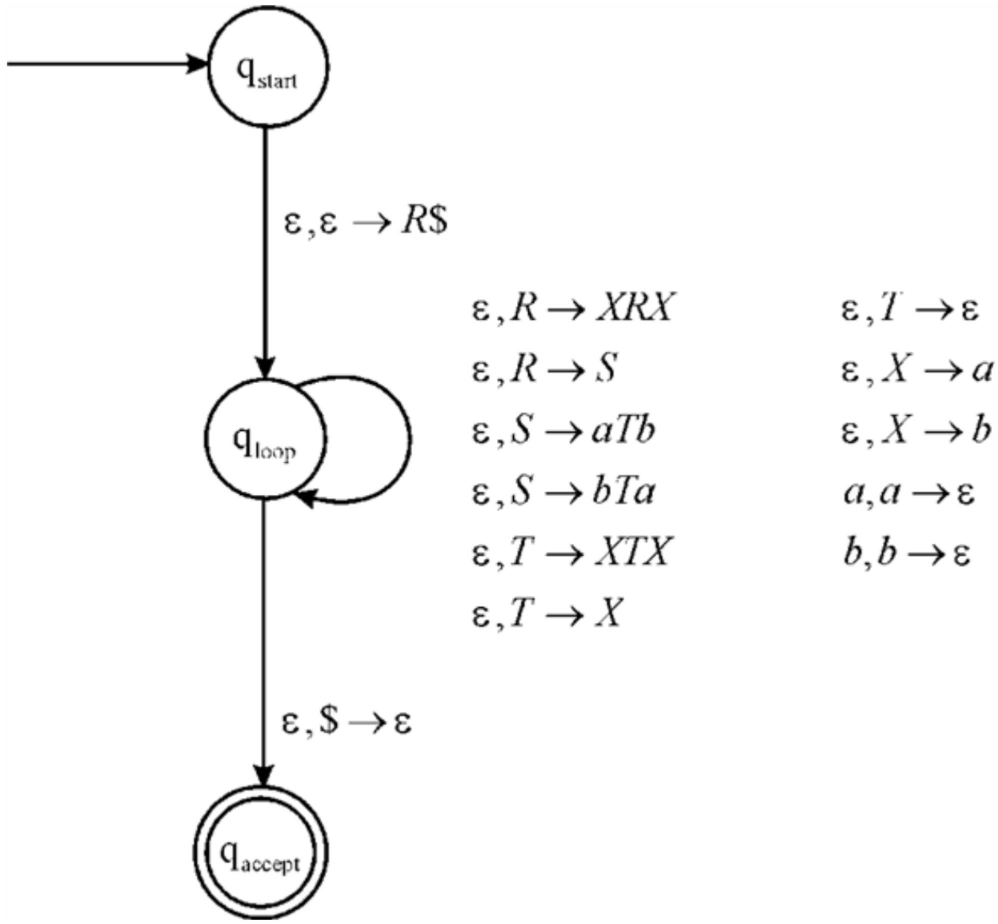
2.11) Convert the CFG to an equivalent PDA

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



2.12) Convert the CFG to an equivalent PDA

$$R \to XRX | S$$
$$S \to aTb | bTa$$
$$T \to XTX | X | \epsilon$$
$$X \to a | b$$



2.14) Convert the following CFG into an equivalent CFG in Chomsky normal form.

$$A \to BAB|B|\epsilon$$
$$B \to 00|\epsilon$$

In Chomsky Normal form every rule is of the form

$A \to BC$
$A \to a$
$S \to \epsilon$, where S is the start symbol

We first add a new start variable S_0 to get

$$\begin{aligned} S_0 &\rightarrow A \\ A &\rightarrow BAB|B|\epsilon \\ B &\rightarrow 00|\epsilon \end{aligned}$$

Next we remove all rules containing ϵ

$$\begin{aligned} S_0 &\rightarrow A|\epsilon \\ A &\rightarrow BAB|BA|AB|A|B|BB \\ B &\rightarrow 00 \end{aligned}$$

Now we remove the unit rules to get the following

$$\begin{aligned} S_0 &\rightarrow BAB|BA|AB|00|BB|\epsilon \\ A &\rightarrow BAB|BA|AB|00|BB \\ B &\rightarrow 00 \end{aligned}$$

Now we can replace the terminals with a new variable U

$$\begin{aligned} S_0 &\rightarrow BAB|BA|AB|UU|BB|\epsilon \\ A &\rightarrow BAB|BA|AB|UU|BB \\ B &\rightarrow UU \\ U &\rightarrow 0 \end{aligned}$$

Now we need to shorten the RHS of the rules to contain only two variables

$$\begin{aligned} S_0 &\rightarrow BC|BA|AB|UU|BB|\epsilon \\ A &\rightarrow BC|BA|AB|UU|BB \\ B &\rightarrow UU \\ U &\rightarrow 0 \\ C &\rightarrow AB \end{aligned}$$

This is the final CFG in Chomsky Normal Form