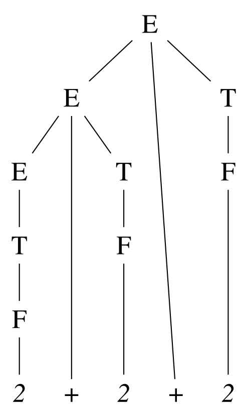
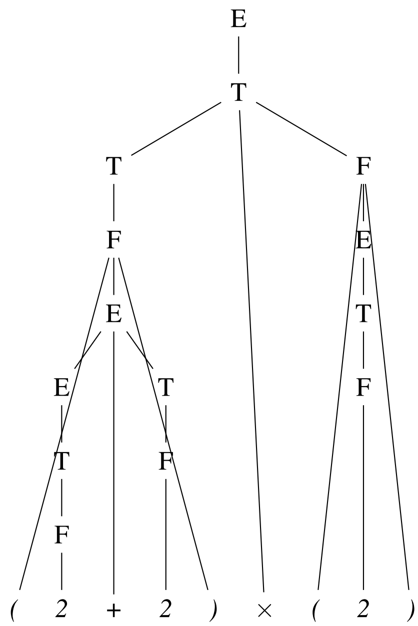
CMSC 303 Introduction to Theory of Computation, VCU
Assignment: 4
Name: Steven Hernandez

1. (a) 

   (b) 

(c) 

2. (a)

   (b)

3. (a)

$$S \to 0|1|0T0|1T1T \to \epsilon|1T|0T$$

Trivially 0 and 1 match. $0T0$ $1T1$ ensure that the first and last symbol are the same before moving past S into T. T simply allows you to add any symbols $\in \Sigma_\epsilon$ recursively within the string obtained above.

(b)

$$S_0 \to 0|1|0S_{odd}|1S_{odd}S_{odd} \to \epsilon|0S_{even}|1S_{even}S_{even} \to 0|1|0S_{odd}|1S_{odd}$$

Think of the labels such that S_{odd} means we have an odd length currently, thus we can only add ϵ of one symbol, which then means we now have an even number of symbols (thus the S_{even}).

(c)

$$S \to \epsilon|0|1|0S0|1S1$$

Unlike a, the only variable is S, this is because each time we recurse, we want to ensure whatever the sub-string contains, it always begins and ends with the same symbol, thus maintaining the palindrome.

(d)

$$S \to S$$

The grammar continues recursively forever. Never reaching only terminals, thus never reaching an accept state.

(e)

$$\begin{array}{lcl} S & \to & X\$C\$X \;|\; C\$X \;|\; X\$C \;|\; C \\ C & \to & 1C1 \;|\; 0C0 \;|\; \$ \;|\; \$X\$ \\ X & \to & \$X \;|\; 1X \;|\; 0X \;|\; \epsilon \end{array}$$

The idea for this grammar is that C always builds a palindrome. Note how from C, we either recursively wrap C with 0 or 1. After which, we can leave \$ in the center.

From the first step, if there are any symbols to the left or right of C, we delimite it with a \$. This keeps the grammar matching the language.

This produces the language when $i = n$ and $j = n + 1$, however does not account for $i = n$ and $j = n + t$ where $t > 1$. So, notice $C \to \$X\$$. This allows the palindrome we were building to have non-palindrome-like items in the middle of this palindrome. Notice though, that these symbols are delimited by $\$$ so that we keep separate the palindrome from earlier.

4. (a)

   (b)

   (c)

   (d)

5.