Problem Set #2

Maxwell Petersen

October 17, 2015

Problem 1

See JFLAP files.

Problem 2

Let there be some string w that is accepted by a CFG G in CNF. The creation of that string following the specific rules to make G to be a CNF will allow there to be a reversal of the string simply by reversing all of the rules of the form $A \to aB$ and $A \to BC$, where A, B, and C are any variable and a is a terminal in Σ, to be $A \to Ba$ and $A \to CB$. This reversal of the rules to generate the string keep G in CNF so G is still Context-Free. Meaning that reversal keeps CFG's closed.

Problem 3

Let there be some string w that is accepted by a CFG G in CNF. With $|w| = 1$ then the amount of derivatives needed will be 1. This is possible with the start variable of G having a terminal rule. For $|w| \geqslant 1$ the format of the rules have it that for every RHS has length at most 2 so for each step that is not a terminal you will have another step to derive until a terminal is reached. Which following that pattern will guarantee that to get to a terminal there will always be $2n - 1$ steps between the start and terminal state.

Problem 4

By observation it can be seen that the A and B LHS variables produce even amount of terminal variables. Excluding the center terminal which will be either an a or a b meaning that there will be a guarantee that the two halves of the generated string will be the same length and that there will be at least one terminal different between the two halves.

Problem 5

Let $G = (V, \Sigma_1, R, S)$ be in FSF. Let there be an NFA $N = (Q, \Sigma_2, \delta, q_0, F)$ be defined as such:

$$
\begin{aligned}
Q &= V \\
\Sigma_2 &= \Sigma_1 \\
\delta &= \{A \to tB \mid \exists R\} \\
q_0 &= S \\
F &= \{A \to \epsilon \mid \exists R\}
\end{aligned}
$$

Problem 6

Let there be a regular language A with $\textsc{Sort}(A)$ being defined as, with $\Sigma = \{a, b\}$
$\textsc{Sort}(A) = \{a^n b^m \mid n \text{ is the number of } a\text{'s and } m \text{ is the number of } b\text{'s for any string } w \in A\}$
Let there be a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes A and let there be a CFG of the following form that will generate $\textsc{Sort}(A)$.
$S \to AB$
$A \to aA \mid \epsilon$
$B \to bB \mid \epsilon$
With S being the start variable. with the proceeding CFG $\textsc{Sort}(A)$, with $\Sigma = \{a, b\}$, is proven to be context-free.

Problem 7

By the definition of $\textsc{Sort}$ found in problem 6 above for a sorted language $\Sigma = \{a, b, c\}$ the new definition of $\textsc{Sort}$ is
$\textsc{Sort}(A) = \{a^n b^m c^o \mid n$ is the number of a's, m is the number of b's, and o is the number of c's for any string $w \in A\}$
seeing how A is regular and we have proven that letting $B = \{a^n b^n c^n \mid n \geqslant 0\}B$ is not context free. By intersecting $\textsc{Sort}(A)$ with B we can see that letting $C = A \cap B$ that C is not a CFG thus $\textsc{Sort}(A)$ is not a CFG as well.