ECE 468 & 573
Problem Set 2: Context-free Grammars, Parsers

1. For the following sub-problems, consider the following context-free grammar:

$$\begin{aligned}
S &\rightarrow AB\$ && (1)\\
A &\rightarrow xAx && (2)\\
A &\rightarrow B && (3)\\
A &\rightarrow \lambda && (4)\\
B &\rightarrow yBy && (5)\\
B &\rightarrow A && (6)\\
B &\rightarrow \lambda && (7)\\
& && (8)
\end{aligned}$$

   (a) What are the terminals and non-terminals of this language?

   (b) Is this a regular language (*i.e.*, could you write a regular expression that generates this language)?

   (c) Show the derivation of the string $xyyxyxxy\$$ starting from S (specify which production you used at each step), and give the parse tree according to that derivation.

   (d) Give the first and follow sets for each of the non-terminals of the grammar.

   (e) What are the predict sets for each production?

   (f) Give the parse table for the grammar. Is this an LL(1) grammar? Why or why not?

2. for the following sub-problems, consider the following grammar:

$$\begin{aligned}
S &\rightarrow AB\$ && (9)\\
A &\rightarrow Ax && (10)\\
A &\rightarrow z && (11)\\
A &\rightarrow x && (12)\\
B &\rightarrow yB && (13)\\
B &\rightarrow w && (14)
\end{aligned}$$

   (a) Explain why this grammar is not LL(k) for any k.

   (b) Build the CFSM for this grammar.

   (c) Build the goto and action tables for this grammar. Is it an LR(0) grammar? Why or why not?

(d) Show the steps taken by the parser when parsing the string: $zxyw\$$. Give the action and show the state stack and remaining input for each step of the parse. Shift actions should be of the form “Shift X” where X is the state you are shifting to, and Reduce actions should be of the form “Reduce R, goto X” where “R” is the rule being used to reduce, and “X” is the state the parser winds up in.

(e) Suppose we change the last production to:

$$B \to x$$

Is the resulting grammar LR(0)? Why or why not?