Alex Irish
Compilers Homework 2
3/2/2018

1. For the following subquestions consider the following grammar:

$$S \to AA\$$$
$$A \to xA$$
$$A \to B$$
$$B \to yB$$
$$B \to \lambda$$

(a) What are the terminals and nonterminals of this language

The terminals are $\$, x, y, \lambda$
The nonterminals are S, A, B

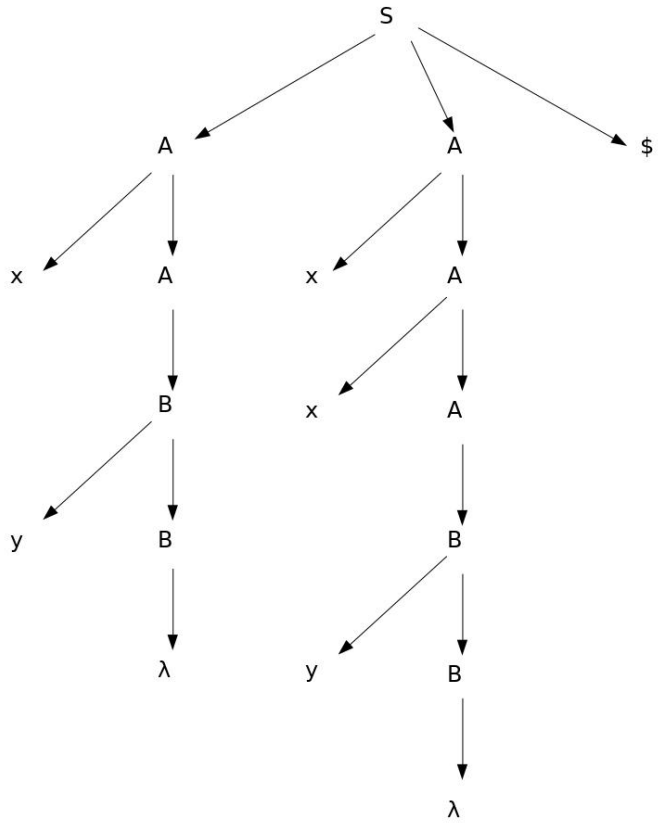
(b) Describe the strings that are generated by this language. Is this a regular language?

This language generates strings that start with 0 or more x's, and ends with 0 or more y's and exactly one \$. It is regular and can be described by the regular expression $x^*y^*x^*y^*\$$.

(c) Show the derivation of the string $xyxxy\$$ starting from S, and give the parse tree according to that derivation.

Derivation	Remaining String	Predict Rule
S	$xyxxy\$$	$Predict(1)$
$AA\$$	$xyxxy\$$	$Predict(2)$
$xAA\$$	$xyxxy\$$	$Match(x)$
$AA\$$	$yxxy\$$	$Predict(3)$
$BA\$$	$yxxy\$$	$Predict(4)$
$yBA\$$	$yxxy\$$	$Match(y)$
$BA\$$	$xxy\$$	$Predict(5)$
$A\$$	$xxy\$$	$Predict(2)$
$xA\$$	$xxy\$$	$Match(x)$
$A\$$	$xy\$$	$Predict(2)$
$xA\$$	$xy\$$	$Match(x)$
$A\$$	$y\$$	$Predict(3)$
$B\$$	$y\$$	$Predict(4)$
$yB\$$	$y\$$	$Match(y)$
$B\$$	$\$$	$Predict(5)$
$\$$	$\$$	$Match(\$)$

Parse Tree



(d) Give the first and follow sets for each nonterminal

$First(S) = \{x, y, \$, \lambda\}$
$First(A) = \{x, y, \lambda\}$
$First(B) = \{y, \lambda\}$

$Follow(S) = \{\$\}$
$Follow(A) = \{x, y, \$\}$
$Follow(B) = \{x, y, \$\}$

(e) Give the predict sets for each production

$Predict(1) = \{x, y, \$\}$
$Predict(2) = \{x\}$
$Predict(3) = \{y, \$\}$
$Predict(4) = \{y\}$
$Predict(5) = \{x, y, \$\}$

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

Left Hand Side	x	y	\$
S	$Predict(1)$	$Predict(1)$	$Predict(1)$
A	$Predict(2)$	$Predict(3)$	$Predict(3)$
B	$Predict(5)$	$Predict(4)$	$Predict(5)$

The grammar is $LL(1)$ because there is at most one predict rule in every place in the table.

2. For the following subquestions consider the following grammar:

$$S \to AB\$$$
$$A \to xB$$
$$A \to xw$$
$$B \to xyA$$
$$B \to z$$

(a) Build a CFSM for this grammar

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

(c) Show the steps taken by the parser when parsing the string: $xzxyxz$. Give the action and show the state stack and remaining input for each step of the parse. Shift actions should be in the form of "Shift X" where X is the state shifting to, and Reduce actions should be in the form "Reduce R, goto X" where R is the rule being use dot reduce and X is the state the parser winds up in.

(d) Suppose we change the last production to:

$$B \to w$$

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