Your name: **Joshua Hoshiko**
Assignment name: Homework 4
Date submitted: 9/25/19
Time spent on assignment: ~3 Hours

"How'd it go?" Overall, I think that the assignment went pretty well. Most of my time was spent thinking of efficient ways to use the stack.

Any remaining questions on the material? I don't think I have any that I can think of.

Who you collaborated with or got help from (if anyone), and what references you consulted beyond the text and course notes. This assignment was completed alone.

If this is an incomplete assignment, what is missing, or not working? Be specific. This assignment is complete.

Additional discussions specified for an individual assignment. None.

Anything else? No, but thank you!!

Theory of Computation
Fall 2019 Context-free Languages
Dr. Gurka

Do not vary the notation or conventions used in class / Sipser.

Do the problems on your own first, then, if you wish, discuss your answers. Do not simply work the problems together. Submission of your assignment is your statement that you followed the collaboration rules. Note that the last problem is strictly on your own.

Part 1: grammars

1. For the following grammars, write the leftmost derivation for the strings given with each. Next to each derivation step, write the number of the rule used. Do not combine steps. (1a to be worked in class, collected problems: 1b, 1c, and 1d)
   a. grammar (3 rules), $\Sigma = \{a, b\}$:
      $S \rightarrow aSbS \mid bSaS \mid \varepsilon$
      strings (3): ab, baab, bbaa
   b. grammar (6 rules), $\Sigma = \{0, 1\}$:
      $S \rightarrow A1B$
      $A \rightarrow 0A \mid \varepsilon$
      $B \rightarrow 0B \mid 1B \mid \varepsilon$
      strings (3): 1, 0011, 01010
   c. grammar (6 rules), $\Sigma = \{E, F, T, +, x, a, (,)\}$:
      $E \rightarrow E + T \mid T$
      $T \rightarrow T x F \mid F$
      $F \rightarrow (E) \mid a$
      strings (3): a, a x a, (a) + a x a,
   d. grammar (4 rules), $\Sigma = \{(,), [,]\}$:
      $S \rightarrow (S) \mid [S] \mid SS \mid \varepsilon$
      strings (3): ( ), [ ] ( ), ([ ])

2. Draw parse trees for nine strings in question one, using their grammars. (2b to be worked in class, collected problems: 2a, 2c, and 2d)

3. Write context-free grammars for the following languages. (3a to be worked in class, collected problems: b, c, d)
   a. strings in reverse alphabetical order, all characters must appear at least once, $\Sigma = \{a, b, c\}$
   b. $1\Sigma^+0$, $\Sigma = \{0, 1\}$

c. strings with either a's and/or b's only, or a's and/or c's only, in any order; strings may be of any length, except ε is not in the language $\Sigma = \{a, b, c\}$,
sample strings for part one of the definition:
~~ε,~~ a, aaa, b, abb, baab, bbabb

d. $w\Sigma\Sigma\Sigma^* w^R$, $\Sigma = \{a, b, c\}$

Part 2: PDAs

Non-collected problems, PDAs: Sipser 2.5, 2.7 (do full PDAs).

4. Create pushdown automata for the following languages. Start each one by briefly outlining your plan in English, explaining how the stack will be used. Also give five good strings in the language. Then give a PDA diagram and formal definition for all parts. (all problems collected)
   a. L3 = $\{ 1^n 2^n 3^m 4^m,\ \Sigma = \{1, 2, 3, 4\} \}$
   b. L1 = $\{ (x \cup y \cup z)^*$, number of x's > number of z's, $\Sigma = \{x, y, z\} \}$
   c. L2 = $\{ a^n b^{n+m} a^m,\ n, m >= 0,\ \Sigma = \{a, b\} \}$
   d. L4 = $\{ xw \mid x^R$ is a prefix of w,
      x & w are elements of $(a \cup b)^+$,
      $|w| > |x| \}$

5. (To be worked strictly on your own: no classmates / friends / tutors / Google / other references). Consider the language $a^n b^n c^n$. Explain what is wrong with the following plan for a PDA to recognize strings in this language. (We know it must be wrong, because this language is not context-free.) Be specific; also give a string that illustrates the problem; state whether it is a rejected good string or an accepted bad string. Hint: create a PDA for considering the system's behavior. Hand in only your explanation and string (not the PDA).

   The plan. In order to match both the 'b's and the 'c's, push extra 'a's. For each 'a' read at the beginning of the string, push two 'a's on the stack. When 'b's are encountered in the string, pop one 'a' and match by consuming one 'b'. When all 'b's are finished, similarly match the remaining 'a's on the stack with 'c's in the string. Note: it is possible to create this PDA, so that is not the problem.

Joshua
Hoshiko

Homework 4

1. a.)

$S \to aSbS$ (1.)
$\to a\varepsilon bS$ (3.)
$\to a\varepsilon b\varepsilon$ (3.)
$\to \boxed{ab}$ remove ε

$S \to bSaS$ (2.)
$\to b\varepsilon aS$ (3.)
$\to b\varepsilon aaSbS$ (1.)
$\to b\varepsilon aa\varepsilon bS$ (3.)
$\to b\varepsilon aa\varepsilon b\varepsilon$ (3.)
$\to \boxed{baab}$ remove ε

$S \to bSaS$ (2.)
$\to bbSaSaS$ (2.)
$\to bb\varepsilon aSaS$ (3.)
$\to bb\varepsilon a\varepsilon aS$ (3.)
$\to bb\varepsilon a\varepsilon a\varepsilon$ (3.)
$\to \boxed{bbaa}$ remove ε

---

b.)

$S \to A1B$ (1.)
$\to \varepsilon 1B$ (3.)
$\to \varepsilon 1\varepsilon$ (6.)
$\to \boxed{1}$ remove ε

$S \to A1B$ (1.)
$\to 0A1B$ (2.)
$\to 00A1B$ (2.)
$\to 00\varepsilon 1B$ (3.)
$\to 00\varepsilon 11B$ (5.)
$\to 00\varepsilon 11\varepsilon$ (6.)
$\to \boxed{0011}$ remove ε

$S \to A1B$ (1.)
$\to 0A1B$ (2.)
$\to 0\varepsilon 1B$ (3.)
$\to 0\varepsilon 10B$ (4.)
$\to 0\varepsilon 101B$ (5.)
$\to 0\varepsilon 1010B$ (4.)
$\to 0\varepsilon 1010\varepsilon$ (6.)
$\to \boxed{01010}$ remove ε

---

c.)

$E \to T$ (2.)
$\to F$ (4.)
$\to \boxed{a}$ (6.)

$E \to T$ (2.)
$\to T \times F$ (3.)
$\to F \times F$ (4.)
$\to a \times F$ (6.)
$\to \boxed{a \times a}$ (6.)

$E \to E + T$ (1.)
$\to T + T$ (2.)
$\to F + T$ (4.)
$\to (E) + T$ (5.)
$\to (T) + T$ (2.)
$\to (F) + T$ (4.)
$\to (a) + T$ (6.)
$\to (a) + T \times F$ (3.)
$\to (a) + F \times F$ (4.)
$\to (a) + a \times F$ (6.)
$\to \boxed{(a) + a \times a}$ (6.)

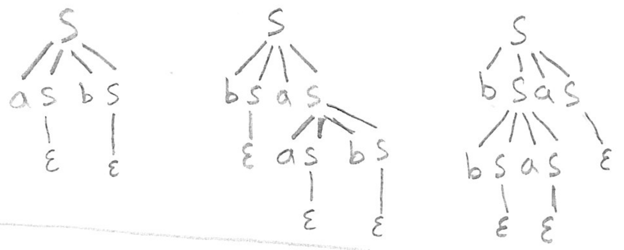
d.)

$S \to (S)$ (1.)
$\to (\varepsilon)$ (4.)
$\to \boxed{(\,)}$ remove ε

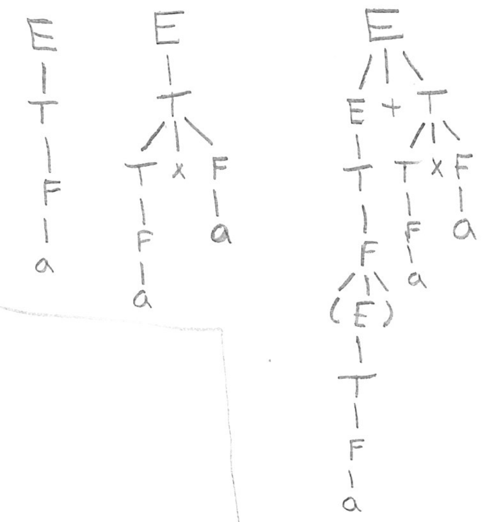
$S \to (S)$ (1.)
$\to ([S])$ (2.)
$\to ([\varepsilon])$ (4.)
$\to \boxed{([\,])}$ remove ε

$S \to SS$ (3.)
$\to [S]S$ (2.)
$\to [\varepsilon]S$ (4.)
$\to [\varepsilon](S)$ (1.)
$\to [\varepsilon](\varepsilon)$ (4.)
$\to \boxed{[\,](\,)}$ remove ε

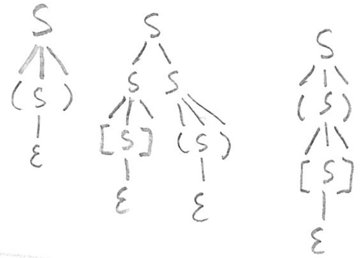
2. a.)



c.)



d.)



Homework 4

3. a.) $S \to CBA$
$C \to cC \mid c$
$B \to bB \mid b$
$A \to aA \mid a$

b.) $S \to 11X0 \mid 10X0$
$X \to 0X \mid 1X \mid \varepsilon$

c.) $S \to X \mid Y$
$X \to aX \mid bX \mid a \mid b$
$Y \to aY \mid cY \mid a \mid c$

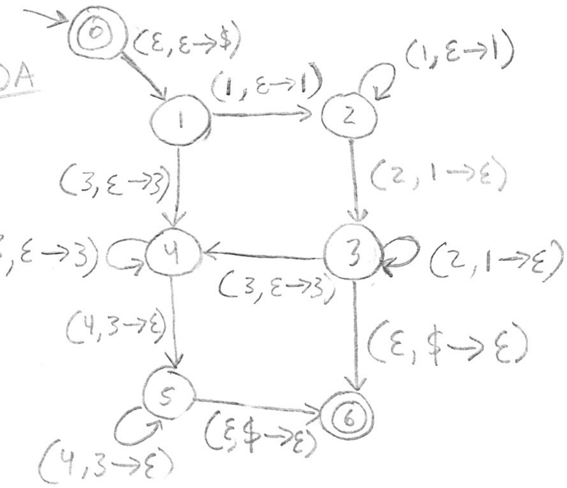
d.) $S \to aSa \mid bSb \mid cSc \mid X$
$X \to aaY \mid abY \mid acY \mid baY \mid$
$bbY \mid bcY \mid caY \mid cbY \mid$
ccY
$Y \to aY \mid bY \mid cY \mid \varepsilon$

(4.) a.) Plan: Have the Stack remember the number of '1's and pop off the Stack for the '2's. The Same Pattern will be used for the '3's and '4's.

good Strings:

112234
111222
123344
34
11223344

PDA



formal definition

$Q = \{0, 1, 2, 3, 4, 5, 6\}$
$\Sigma = \{1, 2, 3, 4\}$
$\Gamma = \{\$, 1, 3\}$
$F = \{0, 6\}$
$S = 0$

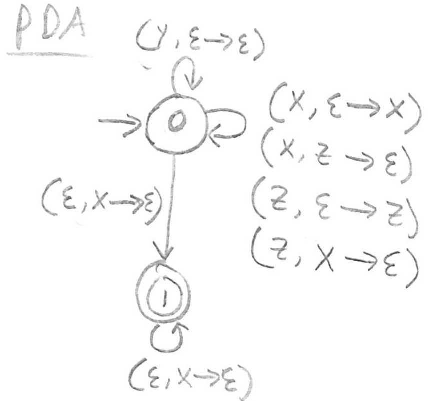
input	1				2				3				4				ε			
Stack	$	1	3	ε	$	1	3	ε	$	1	3	ε	$	1	3	ε	$	1	3	ε
0																				(1,$)
1				(2,1)								(4,3)								
2				(2,1)		(3,ε)														
3						(3,ε)						(4,3)					(6,ε)			
4												(4,3)			(5,ε)					
5															(5,ε)		(6,ε)			
6																				

b.) Plan: use the Stack to track number of 'x's and 'z's whilst also popping off depending on which Character we have a Surplus of. An accepted String will only have 'x's left on the Stack

good Strings

xxyz   xxz
xyyy   xxzyxz
yx   zzxyxx

PDA



Formal definition

$Q = \{0, 1\}$
$\Sigma = \{x, y, z\}$
$\Gamma = \{x, z\}$
$F = \{1\}$
$S = 0$

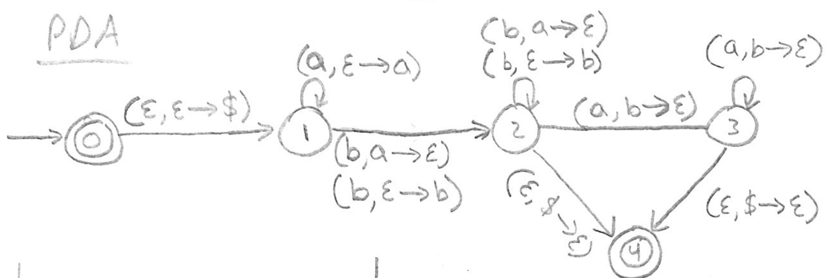
input	x			y			z			ε		
Stack	x	z	ε	x	z	ε	x	z	ε	x	z	ε
0		(0,ε)	(0,x)			(0,ε)	(0,ε)		(0,z)	(1,ε)		
1										(1,ε)		

c.) Plan: Use the Stack to track the number of 'a's at the beginning, then pop off the Stack for half the 'b's, then repeat the process with half the 'b's and the 'a's at the end.

good Strings

aabbbbaa
abbbaa
bbaa
ab
aaabbbba

PDA



Formal definition

$Q = \{0, 1\}$
$\Sigma = \{a, b\}$
$\Gamma = \{\$, a, b\}$
$F = \{0, 4\}$
$S = 0$

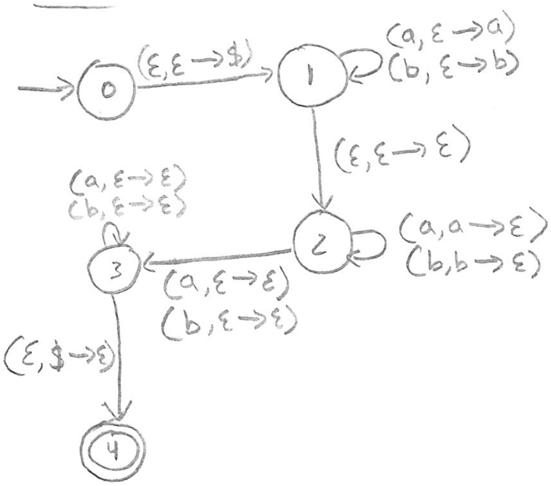
input	a				b				ε			
Stack	a	b	$	ε	a	b	$	ε	a	b	$	ε
0												(1,$)
1				(1,a)	(2,ε)			(2,b)				
2		(3,ε)			(2,ε)			(2,b)			(4,ε)	
3		(3,ε)									(4,ε)	
4												

(4.) d.) Plan: Because there is a palendrome at the beginning, the first half of the PDA will handle the palendrome and second half just has to be checked that there is at least one character on the end of w.

good strings

aaaabab
abbaabba
aab
abaabaaab
bbaabba

PDA



formal definition

$Q = \{0,1,2,3,4\}$
$\Sigma = \{a,b\}$
$\Gamma = \{a,b,\$\}$
$F = \{4\}$
$S = 0$

input	a				b				ε			
Stack	a	b	$	ε	a	b	$	ε	a	b	$	ε
0												(1, $)
1				(1, a)				(1, b)				(2, ε)
2	(2, ε)			(3, ε)		(2, ε)		(3, ε)				
3				(3, ε)				(3, ε)			(4, ε)	
4												

(5.) The issue with the plan is that the 'b's and 'c's only have to collectively equal the same number of 'a's times two. For example, "aabccc" would be accepted because bccc is length four, but the string is not in the language. This means that the issue is that the plan can't check for an even distribution of 'b's and 'c's

bad strings that are accepted

aabccc

$\underbrace{\underbrace{\text{aaa bbbb}}_{\text{length 4}}\underbrace{\text{cc}}_{\text{length 2}}}_{\text{length } 6 = 2*3}$