CS 5000 Homework 5

Sally Devitry (A01980316)

Problem 1

Give a CFG for each of the following languages:
1. $L1 = a^b$;
2. $L2 = \{x \in \{0,1\}^* \mid$ the number of 0's in x is divisible by 3$\}$;
3. $L3 = a^n b^n, n = 2k, k \in N$;
4. $L4 = a^n b^m c^m d^n, n, m \in N$.

Solution

1. $G = (V, \Sigma, S, P)$ with set of variables $V = a, b$, set of terminals $\Sigma = \{S, A\}$; and rules
$S \rightarrow aS \mid aA \mid \epsilon$
$A \rightarrow bA \mid \epsilon$

2. $G = (V, \Sigma, S, P)$ with set of variables $V = 0, 1$, set of terminals $\Sigma = \{S, A, B\}$; and rules
$S \rightarrow 1S \mid 0A \mid \epsilon$
$A \rightarrow 1A \mid 0B$
$B \rightarrow 1B \mid 0S$

3. $G = (V, \Sigma, S, P)$ with set of variables $V = a, b$, set of terminals $\Sigma = \{S\}$; and rules
$S \rightarrow aaSbb \mid \epsilon$

4. $G = (V, \Sigma, S, P)$ with set of variables $V = a, b, c, d$, set of terminals $\Sigma = \{S, A\}$; and rules
$S \rightarrow aSd \mid A$
$A \rightarrow bAc \mid bc$

Problem 2

Construct a stack machine for each language in Problem 1 by giving its transition table.

Solution

Table for L_1

read	pop	push
ϵ	S	ϵ
ϵ	S	aS
ϵ	S	aA
ϵ	A	ϵ
ϵ	A	bA
a	a	ϵ
b	b	ϵ

Table for L_2

read	pop	push
ϵ	S	ϵ
ϵ	S	$1S$
ϵ	S	$0A$
ϵ	A	$1A$
ϵ	A	$0B$
ϵ	B	$1B$
ϵ	B	$0S$
1	1	ϵ
0	0	ϵ

Table for L_3

read	pop	push
ϵ	S	ϵ
ϵ	S	$aaSbb$
a	a	ϵ
b	b	ϵ

Table for L_4

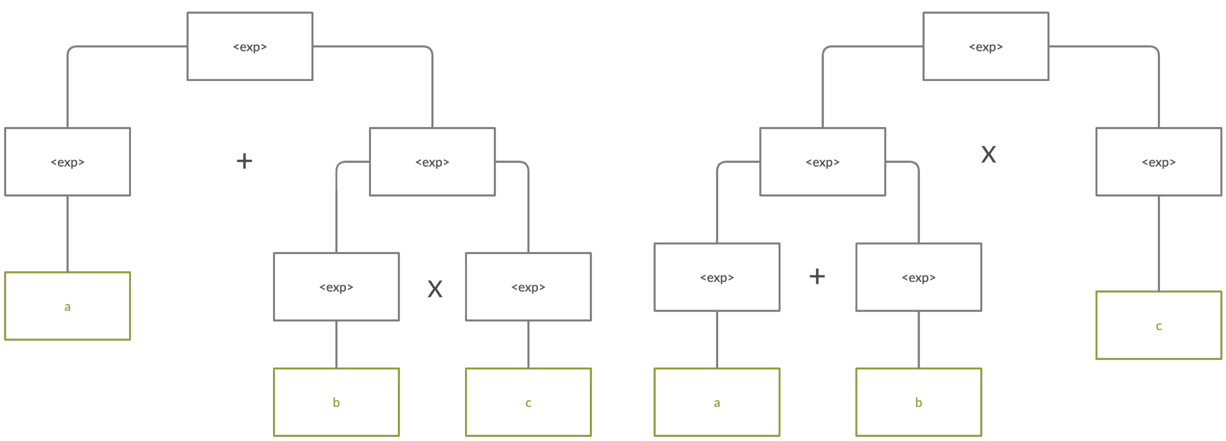
read	pop	push
ϵ	S	aSd
ϵ	S	A
ϵ	A	bAc
ϵ	A	bc
a	a	ϵ
b	b	ϵ
c	c	ϵ
d	d	ϵ

Problem 3

A grammar G is ambiguous if it produces several (more than 1) derivation trees for the same string in $L(G)$. Consider the following grammar:
1. $<$ `exp` $>\rightarrow<$ `exp` $>+<$ `exp` $>$;
2. $<$ `exp` $>\rightarrow<$ `exp` $>*<$ `exp` $>$;
3. $<$ `exp` $>\rightarrow$ ($<$ `exp` $>$);
4. $<$ `exp` $>\rightarrow$ `a`;
5. $<$ `exp` $>\rightarrow$ `b`;
6. $<$ `exp` $>\rightarrow$ `c`;
Show that G is ambiguous.

Solution

The grammar, G, is ambiguous because there are two derivation trees that can be given for a string in the language. Take the example string, $a + b * c$. The two unique derivation trees are shown below.



Problem 4

Recall the following grammar we investigated in lecture 8:
1. $S \rightarrow aSa$;
2. $S \rightarrow bSb$;
3. $S \rightarrow \epsilon$.
Show that $L(G) = \{xxR \mid x \in \{a, b\}\}$.

Solution

$\{xx^R\}$ is the set of strings that can be formed by taking any string from $(a, b)^*$ and appending the same string in reverse.

For example, the string, $abba$, is in the language (x='ab'). A derivation of $abba$ can be given by:
$S \to aSa$ (rule 1)
$\to abSba$ (rule 2)
$\to ab\epsilon ba$ (rule 3)
$\to abba$.
As we can see, everytime we use one of the first two rules, we add a symbol to the end of the first half, and the same symbol at the start of the second half of the string. Thus, the second half is always the reverse of the first half: $L(G) = \{xx^R \mid x \in \{a,b\}^*\}$.

Problem 5

Convert the following grammar into CNF.
1. $S \to ABA \mid BAS$;
2. $A \to a$;
3. $B \to b$.

Solution

Production 1 is the only rule not yet in CNF. The start symbol appears on the R.H.S., so we must create a new start symbol S' and add the production $S' \to S$.
CNF does not allow unit productions, so the right side of $S' \to S$ should be replaced by the right side of S.
CNF only allows for two nonterminals on the R.H.S, so we must add production $C \to BA$, so that the production for S becomes $S \to AC \mid CS$

The final grammar in CNF is:
1. $S' \to AC \mid CS$;
2. $S \to AC \mid CS$;
3. $C \to BA$;
4. $A \to a$;
5. $B \to b$;

Problem 6

Consider the following grammar G.
1. $S \to AB \mid BC$;
2. $A \to BA \mid a$;
3. $B \to CC \mid b$;
4. $C \to AB \mid a$;

Let $x = baab$. Construct the table D[s, l] for the Cocke-Younger-Kasami algorithm and determine if $x \in L(G)$.

Solution

To begin, we construct a 4 by 4 table as the length of the string is 4. One side of the table represents the 'starts at' position, and one represents the length of the substring. Each cell in the following table was calculated by looking at the grammar to see what could be derived from a string of length l and starting at position S.

S (starts at)

l (length)	1	2	3	4
1	{B}	{A, C}	{A, C}	{B}
2	{S, A}	{B}	{S, C}	
3	{ }	{B}		
4	{ }			

Because cell [1, 4] does not have the start state, S, we can say that the string, *baab*, is not accepted by the language, $baab \notin L(G)$