1

Formal Languages and Compilers

Name: ______________________

1 – Verify the equivalence of the following regular expressions :

(0 | 11*0)* 11*

0*1 (00*1 | 1)*

2 – Find a pushdown automaton accepting the language: $\{ a^m b^{m+n} a^n \mid m >= 0 , n >= 0 \}$.

3 – Find the predictive parsing table for the following grammar:

$S \rightarrow AB$
$S \rightarrow B$
$A \rightarrow aA$
$A \rightarrow \varepsilon$
$B \rightarrow bB$
$B \rightarrow \varepsilon$

4 – Find the LR(1) parsing table for the following grammar:

1) $S \rightarrow AaAb$
2) $S \rightarrow BbBa$
3) $A \rightarrow \varepsilon$
4) $B \rightarrow \varepsilon$

2

Formal Languages and Compilers

Name: ______________________

1 – Find the minimum state automaton accepting the union of the languages denoted by the following regular expressions :

(0*1 | 10*)* 0

(0 | 11)* 01

2 – Find a pushdown automaton accepting the language: $\{ b^{2m} a^{n+m+1} b^{2n} \mid m >= 0 , n >= 0 \}$.

3 – Find the LALR(1) parsing table for the following grammar:

$S \rightarrow S\ S + \mid x$

4 – Transform the following SDT so that it can be implemented by a bottom-up parser.
Then indicate which string will be printed when the input is “ccaba”:

S → c{print "x"} S S
S → {print "y"} a
S → b {print "z"}

3

Formal Languages and Compilers

Name: ________________________

1 – Find the minimum state automaton accepting the language denoted by the regular expression :

$$(a\, b^* \mid b^*\, a)^*$$

2 – Find a pushdown automaton accepting the language: $\{ (a^n b^n)^m c^m \mid m >= 0 , n >= 0 \}$.

3 – Find the LL(1) parsing table for the following grammar:

$S \rightarrow (L) \mid a$
$L \rightarrow L , S \mid S$

4 – Transform the following translation scheme so that the inherited attributes are defined by *copy rules*:

A → a B C { A.a := f (B.b) }
B → b C { A.b := g (C.c) } A { B.b := k (b.x) }
C → c B { A.b := h (C.c) } A { C.c := k (c.x) }

4

Formal Languages and Compilers

Name : ______________________

1 – Find the minimum automaton accepting the language denoted by:

$$(a\, b^* \mid b^*\, a)^*$$

2 – Find a grammar equivalent to the following one, having only useful symbols:

$S \rightarrow a\ A\ a \mid b\ C\ B\ b$
$A \rightarrow a\ S\ a \mid b\ A \mid b$
$B \rightarrow b\ S\ B\ b \mid a\ B\ C\ b$
$C \rightarrow a\ B\ C\ a \mid a\ A\ S \mid a$

3 – Produce the LR(1) parsing table for the following grammar :

$E \rightarrow id \mid id\ (\ E\ ) \mid E + id \mid \&\ E$

4 – Assuming that the following SDT is implemented by a bottom-up parser, indicate the string that it will print when the input is "baba" :

S → aS {print "x"}
S → {print "y"} bS
S → {print "w"} a
S → b {print "z"}

5

Formal Languages and Compilers

1 – Verify the equivalence between the following grammars:

$S \rightarrow A\ a \mid A\ b \mid S\ b \mid a \mid b$	$S \rightarrow a\ A \mid b\ A \mid a \mid b$
$A \rightarrow S\ a$	$A \rightarrow a\ S \mid b\ A \mid b$

2 – Find a pushdown automaton accepting the strings in $\{\ a^n\ b^m \mid 2\ n <= m <= 3\ n\ \}$

3 – Produce an L-attributed translation scheme equivalent to the following syntax-directed-definition:

$A \rightarrow B\ a\ C\ D$	$A.x = f(B.y, C.z)$
	$C.z = g(B.x)$
	$B.y = h(A.y)$
	$D.z = k(A.w)$

4 – Eliminate left recursion from the following grammar and produce its predictive parsing table:

$S \rightarrow Aa \mid Bc$
$A \rightarrow aB \mid Bb$
$B \rightarrow Sc \mid a$

6

Formal Languages and Compilers

Name : ______________________

1 – Find a regular expression denoting the complement of the language represented by:

$(1|10)^*$

2 – Find a pushdown automaton accepting the language: $\{ (10)^n\ 11\ (01)^n \mid n >= 0 \}$.

3 – Produce the LR(1) parsing table for the following grammar :

$S \rightarrow AaAb \mid BbBa$
$A \rightarrow bA \mid \varepsilon$
$B \rightarrow \varepsilon$

4 – Assuming that the following SDT is implemented by a bottom-up parser, indicate the string that it will print when the input is "ababb" :

S → aS {print "x"}
S → bS {print "y"}
S → a {print "w"}
S → b {print "z"}

7

Formal Languages and Compilers

Name : ______________________

1 – Verify the equivalence of the following grammars:

$S \to A\ b \mid C\ b \mid \varepsilon$
$A \to S\ a \mid a$
$B \to A\ a \mid D\ a$
$C \to B\ b \mid D\ b$
$D \to C\ a$

$S \to a\ A \mid \varepsilon$
$A \to a\ B \mid b\ S \mid b$
$B \to b\ A \mid b\ S \mid b$

2 – Find a pushdown automaton accepting all the strings in $\{ a , b , c \}^*$ with just one "c" and an equal number of "a" and "b".

3 – Produce the predictive parsing table for the following grammar :

$S \to AB \mid B$
$A \to aA \mid \varepsilon$
$B \to bB \mid \varepsilon$

4 – Transform the following SDT so that it can be implemented by a bottom-up parser, and then indicate which string it will print when the input is "cabcbb":

S → c {print "z"} S S
S → a {print "x"} S
S → b {print "y"}

8

Formal Languages and Compilers

Name : ______________________

1 – Verify the equivalence of the following regular expressions:

(0 | 01) (01)*
0 | (0 | 00) (10)* 1

2 – Find a pushdown automaton accepting the language: $\{ (a^n\ b^n)^m\ c^m \mid m >= 0 , n >= 0 \}$.

3 – Produce the predictive parsing table for the following grammar :

S → AB | B
A → Ba | a
B → Sb | b

4 – Transform the following SDT so that it can be implemented by a bottom-up parser, and then indicate which string it will print when the input is “cacba” :

S → cS {print "x"} S
S → a {print "y"}
S → b {print "z"}

9

Formal Languagesi and Compilers

1 – Find the minimum-state DFA equivalent to the following grammar

$S \to A\,0 \mid B\,0 \mid \varepsilon$
$A \to B\,1 \mid 1$
$B \to B\,1 \mid S\,0 \mid 0$

2 – From the following grammar:

$S \to 0\,S\,0 \mid 1C1 \mid BB$
$A \to 0\,C\,1 \mid 0$
$B \to 1\,B \mid B\,A\,C$
$C \to 0\,B \mid 1\,S \mid 1$

eliminate useless symbols

3 – Find a pushdown automaton accepting the strings in $\{0, 1\}^*$ having the same number of 0 and 1

4 – Find the LR(0) parsing table for the following grammar:

$S \to Ab \mid aBa$
$A \to aBA \mid Sa \mid b$
$B \to bAb \mid a$

(10)

Formal Languagesi and Compilers

1 – Find a minimum-state DFA equivalent to the following grammar and then the regular expression for the accepted language

$S \to A\ 0 \mid B\ 0$
$A \to B\ 1 \mid 1$
$B \to B\ 1 \mid S\ 0 \mid 0$

2 – From the following grammar:

$S \to 0\ S\ 0 \mid 1B1 \mid BB$
$A \to 0\ C\ 1 \mid 0$
$B \to 1\ B \mid S\ 0 \mid \varepsilon$
$C \to 0\ A \mid 1$

eliminate useless symbols and ε-productions

3 – Find the SLR parsing table for the following grammar:

$S \to Ab \mid aBa \mid \varepsilon$
$A \to aBA \mid Sa$
$B \to bAb \mid a$

4 – Make the inherited attributes in the following translation scheme be defined by *copy rules*:

$A \to a\ B\ C\ \{ A.a := f(B.b) \}$
$B \to b\ C\ \{ A.b := g(C.c) \}\ A\ \{ B.b := k(b.x) \}$
$C \to c\ B\ \{ A.b := h(C.c) \}\ A\ \{ C.c := k(c.x) \}$

11

Formal Languages and Compilers

1 – Find a left-regular grammar that generates the complement of the language denoted by the following expression:

$$(a\, b\, (a\, b^* \mid b)\, a)^*$$

2 - Find a pushdown automaton accepting the language: $\{ a^n b^m c^k \mid n = 2m \textbf{ or } m = 2k \}$.

3 – Tell if the following grammar is LL(1):

$S \rightarrow (L) \mid a$
$L \rightarrow L , S \mid S$

4 – Tell if the following grammar is LALR(1):

$S \rightarrow A\, a \mid b\, A\, c \mid d\, c \mid b\, d\, a$
$A \rightarrow d$