Faculty of Computers and Information
Department: Computer Science
1st, Semester Final Exam.
Date: 9/1/2023

Menoufia University

Subject: Compiler Design
Year: 2022/2023
Time allowed: 3 Hours
Full Mark: 60

Question-1) (20 marks):

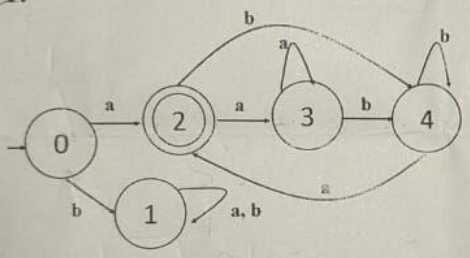
A. What is the difference between the compiler and the Interpreter? List the main Implementation Techniques of Compiler.

B. Describe the languages denoted by the following regular expressions:
1. 1(1+0)*1
2. 1*01*01*01*

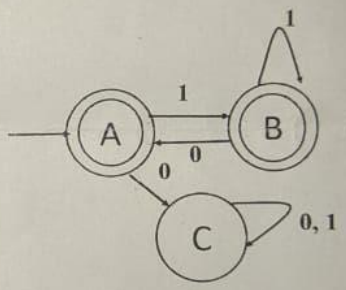
C. Show the finite state machine in **graph form** for each of the following languages (in each case the input alphabet is {a,b}):
1. All strings which **contain exactly one a**.
2. All strings which **end with ab**.

D. Write the regular expression for the following finite state machines:

1.



2.



Question-2) (20 marks):

A. Consider the following grammar: $S \rightarrow aSbS \mid bSaS \mid \varepsilon$

1. Is the grammar ambiguous or unambiguous? Justify your answer.

B. Show three different derivations using the following grammars with starting nonterminal S

$S \rightarrow 0\,S \mid 1A$
$A \rightarrow 1\,S \mid 0$

Faculty of Computers and Information
Department: Computer Science
2nd, Semester Final Exam.
Date: 14/6/2022

Menoufia University

Subject: Compiler Design
Year: 2021/2022
Time allowed: 3 Hours
Full Mark: 70

Question-1) (15 marks):

A. Consider the following grammar: $S \rightarrow aSbS \mid bSaS \mid \varepsilon$

1. Is the grammar ambiguous or unambiguous? Justify your answer?

B. Show one-state pushdown machine and recursive descent parser (**Only S()**) for the following grammar:

$S \rightarrow 0S1 \mid 1$

C. Show the sequence of stacks for the pushdown machine you created above in Q1(B) for this input string, **010010**.

Question-2) (15 marks):

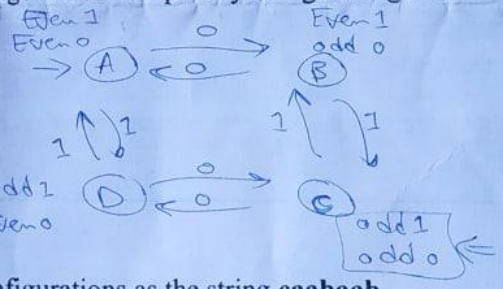
A. Describe the languages denoted by the following regular expressions:

1. a(a+b)*a
2. a*ba*ba*ba*

B. Write the regular expressions for each of the following languages (in each case the input alphabet is {a,b}):

1. Strings containing an odd number of the character b
2. Strings containing the substring abb

C. Show a finite state machine for the language which accepts any string having an odd number of 1's and an odd number of 0's.



Question-3) (20 marks):

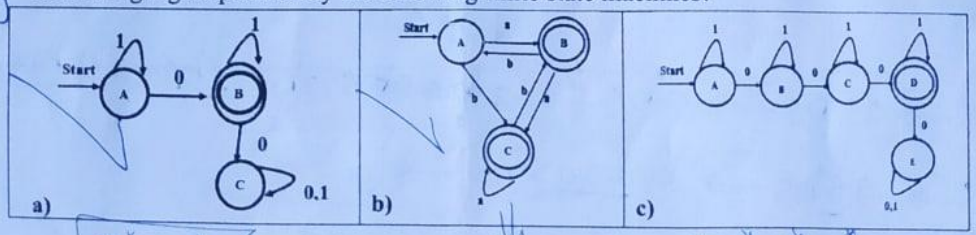
A. Is the following grammar is **LL(1)?** Why?

$S \rightarrow Aa$
$A \rightarrow BD$
$B \rightarrow b \mid \varepsilon$
$D \rightarrow d \mid \varepsilon$

B. Show the sequence of stack and input configurations as the string **caabaab**

$S \rightarrow S\,a\,B \mid c$
$B \rightarrow ab$

C. For each of the following finite state machines, write the regular expressions to specify each of the languages specified by the following finite state machines?



Question-4) (20 marks):

A. Find the selection set for each rule of the following grammar?

$S \rightarrow a\ b\ S\ d$
$S \rightarrow b\ a\ S\ d$
$S \rightarrow d$

B. Show three different derivations using the following grammars with starting nonterminal S.

$S \rightarrow 0\ S \mid 1A$
$A \rightarrow 1\ S \mid 0$

C. Show the sequence of stack, input, action, and goto configurations for the input **(var*var)+var** with a shift reduce parser, using the following grammar.

1. Expr → Expr + Term
2. Expr → Term
3. Term → Term * Factor
4. Term → Factor
5. Factor → (Expr)
6. Factor → var

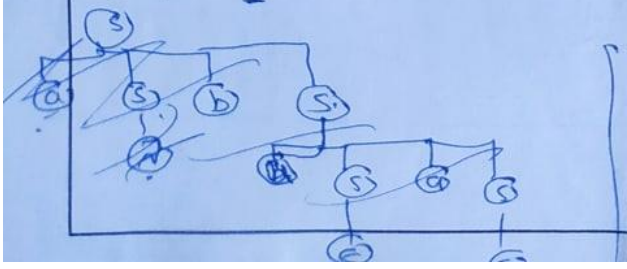
	Action Table					
	+	*	(	)	var	↵
∇			shift (		shift var	
Expr 1	shift +					Accept
Term1	reduce 1	shift *		reduce 1		reduce 1
Factor3	reduce 3	reduce 3		reduce 3		reduce 3
(			shift (		shift var	
Expr5	shift +			shift)		
)	reduce 5	reduce 5		reduce 5		reduce 5
+			shift (		shift var	
Term2	reduce 2	shift *		reduce 2		reduce 2
*****			shift (		shift var	
Factor4	reduce 4	reduce 4		reduce 4		reduce 4
var	reduce 6	reduce 6		reduce 6		reduce 6

	GoTO Table		
	Expr	**Term**	**Factor**
∇	push Expr1	push Term2	push Factor4
Expr1			
Term1			
Factor3			
(	push Expr5	push Term2	push Factor4
Expr5			
)			
+		push Term1	push Factor4
Term2			
*****			push Factor3
Factor4			
Var			

1- S → asbs
2- S → bsas
3- S → ε

S → asbs → abs → abasbs → abab

S → asbs → absasbs → abab



With my best wishes
Dr. Ibrahim A. Elgendy

Faculty of Computers and Information
Department: Computer Science
2nd, Semester Final Exam.
Date: 9/6/2022

Menoufia University

Subject: Compiler Design
Year: 2021/2022
Time allowed: 3 Hours
Full Mark: 80

Question-1) (20 marks):

A. Describe the languages denoted by the following regular expressions:
1. a(a+b)*a
2. a*ba*ba*ba*

B. For each of the above regular expressions, **list four** strings which are in its language.

C. Write the regular expressions for each of the following languages (in each case the input alphabet is {a,b}):
1. Strings containing an odd number of the character b
2. Strings containing the substring abb

D. Show a finite state machine for the language which accepts any string having an odd number of 1's and an odd number of 0's.

Question-2) (20 marks):

A. Consider the following grammar: $S \rightarrow S+S \mid S*S \mid a$
1. Give a parse tree for the string a+a*a.
2. Is the grammar ambiguous or unambiguous? Justify your answer?

B. Show one-state pushdown machine and recursive descent parser (**Only S()**) for the following grammar:

$S \rightarrow 0S1$

$S \rightarrow 1$

C. Show the sequence of stacks for the pushdown machine you created above in Q2(B) for this input string, **010010**.

Question-3) (20 marks):

A. Consider the following grammar:

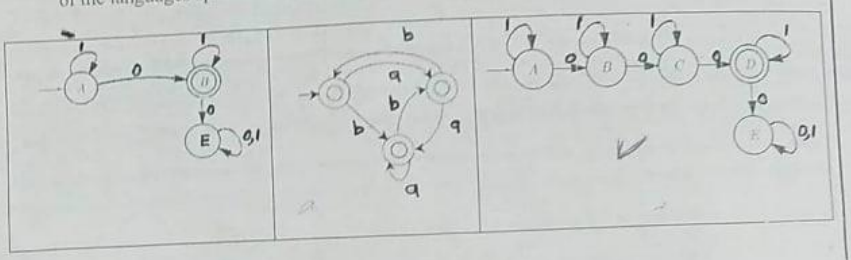
S → SaB
S → b
B → a
B → Bba

Is this grammar LL(1)? Why?

B. Show the sequence of stack and input configurations as the string **caab**

S → S a B
S → c
B → ab

C. For each of the following finite state machines, write the regular expressions to specify each of the languages specified by the following finite state machines?



Question-4) (20 marks):

A. Find the selection set for each rule of the following grammar?

$$S \rightarrow a\ b\ S\ d$$
$$S \rightarrow b\ a\ S\ d$$
$$S \rightarrow d$$

B. Show the sequence of stack, input, action, and goto configurations for the input **(var+var)*var** with a shift reduce parser, using the following grammar.

1. Expr → Expr + Term
2. Expr → Term
3. Term → Term * Factor
4. Term → Factor
5. Factor → (Expr)
6. Factor → var

With my best wishes
Dr. Ibrahim A. Elgendy

Question 2 (10 marks):

1. Consider the following grammar:

$$S \rightarrow SAB \mid \lambda$$
$$A \rightarrow AaB \mid a$$
$$B \rightarrow AS \mid b$$

Is the grammar ambiguous or unambiguous? Justify your answer.

2. Show a finite state machine in table form and regular expressions for the language, where the alphabet is {0, 1}
   a) Strings containing an even number of **0** and ending with **0**.
   b) Strings containing **0100**.

a) **FSM**	b) **FSM**
RE	**RE**

3. Construct the finite state machine in graph form which specifies the same language as each of the following regular expressions. The alphabet is the binary digits {a, b, c}.
   a) c(a+b)*c
   b) (bb)*(aa)*cc

a)	b)

With my best wishes
Dr. Ibrahim A. Elgendy

Faculty of Computers and Information Department: Computer Science 2nd, Semester Mid-Term Exam. Date: 17/11/2022	Menoufia University	Subject: Compiler Design Year: 2022/2023 Time allowed: 50 Minutes Full Mark: 20

Name:	Section:

Answer the following questions (Only one answer for each question)

Question 1 (10 marks):

1. What's the difference between Bootstrapping and Cross Compiling with an example for each one?

Bootstrapping	Cross Compiling

2. Show a one-state pushdown machine and recursive descent parser for the following grammar:

$$S \rightarrow aSbS \mid bSaS \mid \lambda$$

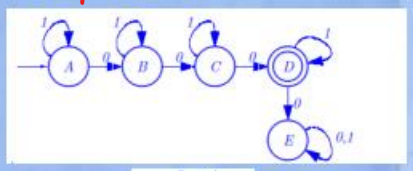
One-state Pushdown Machine	Recursive Descent Parser

3. Show the sequence of stacks for the pushdown machine you created above for this input string, **aababbba**.

Question-2 (10 marks):

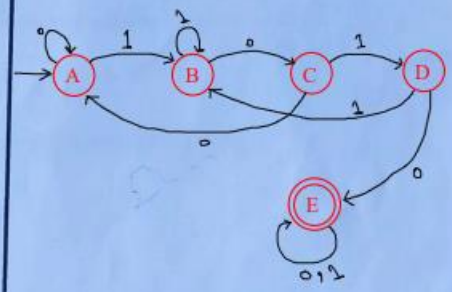
1. Show a finite state machine in either state graph or table form for the language
   a) Strings containing exactly three zeros.
   b) Strings containing 1010.

a) Strings containing exactly three zeros



	0	1
A	B	A
B	C	B
C	D	C
*D	E	D
E	E	E

c) Strings containing 1010



2. Write regular expressions for each description. The alphabet is the binary digits {0, 1}.
   a) All strings which contain three sequential ones.
   b) All strings which contain exactly one 0.
   c) All strings which contain at least three zeros.
   d) All strings which contain an even number of 1s and any number of 0s.

a) (0+1)*111(0+1)*	b) 1*01*
c) (0+1)*0(0+1)*0(0+1)*0(0+1)*	d) 0*(10*10*)*

3. Describe the languages denoted by the following regular expressions
   a) a(a+b)*a
   b) a*ba*ba*ba*

a) Strings start by a and end by a لازم كل string ال يطلع من هذه ال language يبتدي بحرف ال a وينتهي بحرف ال a aa , aaa , aba , aaba , aababa	b) Strings containing at least three b string يحتوي علي الاقل ثلاثه من ال b bbb , abbb , abababa , aabaabaabaa

اللهم صلي وسلم وبارك علي محمد في الملأ الأعلي الي يوم الدين

With my best wishes
Dr. Ibrahim A. Elgendy

Faculty of Computers and Information — Menoufia University — Subject: Compiler Design

Department: Computer Science — Year: 2021/2022

2nd, Semester Mid-Term Exam. — Time allowed: 50 Minutes

Date: 9/4/2022 — Full Mark: 20

Name:	Section:

Answer the following questions:

Question-1 (10 marks):

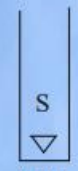
1. Show one-state pushdown machine and recursive descent parser (**Only S()**) for the following grammar:

$S \rightarrow 0S1$

$S \rightarrow 1$

One-state Pushdown Machine

	0	1	↵
S	Rep(1S0) Retain	Rep(1) Retain	Reject
0	pop advance	Reject	Reject
1	Reject	pop advance	Reject
▽	Reject	Reject	Accept

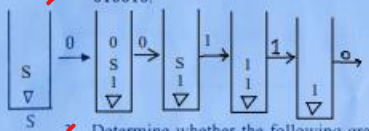


initial

Recursive Descent Parser

```
void S() {
  if(inp=='0'){
     inp = getInp() ;
     S() ;
     if(inp=='1')
        inp = getInp() ;
     else reject() ; }
  else if(inp=='1')
     inp = getInp() ;
  else  reject() ;
}
```

2. Show the sequence of stacks for the pushdown machine you created above for this input string, **010010**.



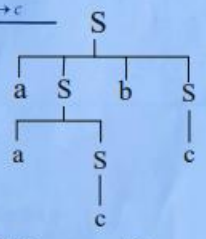
Reject

3. Determine whether the following grammar is ambiguous. If so, show two different derivation trees for the same string of terminals, and show a left-most derivation corresponding to each tree.

$S \rightarrow aSbS$

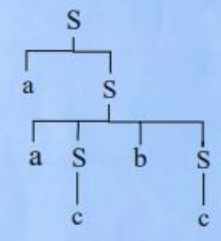
$S \rightarrow aS$

$S \rightarrow c$



left most derivation

S==> aSbS==>aaSbS==>>aacbS==>>aacbc



left most derivation

S==>aS==>aaSbS==>aacbS==>aacbc

this is Ambiguous

Question-3) (20 marks):

A. Show a one-state pushdown machine and recursive descent parser (**Only S()**) for the following grammar:

$$S \rightarrow 1S0 \mid 0$$

B. Show the sequence of stacks for the pushdown machine you created above in Q3(A) for this input string, **11000**.

Question-4) (20 marks):

A. Is the following grammar is **LL(1)?** Why?

$S \rightarrow Aa$

$A \rightarrow BD$

$B \rightarrow b \mid \varepsilon$

$D \rightarrow d \mid \varepsilon$

B. Show the sequence of stack, input, action, and goto configurations for the input **(var*var)+(var*var)** with a shift reduce parser, using the following grammar.

1. Expr → Expr + Term
2. Expr → Term
3. Term → Term * Factor
4. Term → Factor
5. Factor → (Expr)
6. Factor → var

	Action Table					
	+	*	(	)	var	↵
∇			shift (		shift var	
Expr 1	shift +					Accept
Term1	reduce 1	shift *		reduce 1		reduce 1
Factor3	reduce 3	reduce 3		reduce 3		reduce 3
(			shift (		shift var	
Expr5	shift +			shift)		
)	reduce 5	reduce 5		reduce 5		reduce 5
+			shift (		shift var	
Term2	reduce 2	shift *		reduce 2		reduce 2
*****			shift (		shift var	
Factor4	reduce 4	reduce 4		reduce 4		reduce 4
var	reduce 6	reduce 6		reduce 6		reduce 6

	GoTO Table		
	Expr	**Term**	**Factor**
∇	push Expr1	push Term2	push Factor4
Expr1			
Term1			
Factor3			
(	push Expr5	push Term2	push Factor4
Expr5			
)			
+		push Term1	push Factor4
Term2			
*****			push Factor3
Factor4			
Var			

With my best wishes
Dr. Ibrahim A. Elgendy