2nd Term 2020-2021

Department of Computer Science

King Saud University

St. Name:

St. Number

Q1) (14 marks) Put a circle around the symbol of the best answer for each of the following

1) Which of the following features must be available in a programming language to be suitable for writing embedded systems
   - [✓ a.] Provide constructs for low-level control over hardware.
   - b. Use floating point representation
   - c. Have features to analyse data
   - d. All of the above.

2) An orthogonal programming language
   - a. Has a relatively small set of primitive constructs that can be combined in a relatively small number of ways to get the desired results.
   - b. Has a fewer number of exceptions because every possible combination is legal.
   - c. Is good with respect to reliability
   - [✓ d.] All of the above

3) Which of the following features a programming language need to have in order to be good with respect to reliability
   - a. Data types
   - b. Support for abstraction
   - c. Exception handling
   - [✓ d.] all of the above

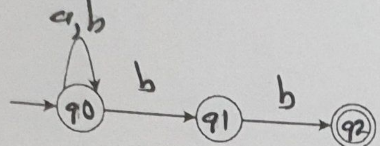
4) Having a small number of manageable features and constructs, makes a programming language good with respect to
   - a. Readability
   - b. Writability
   - c. Reliability
   - [✓ d.] All of the above

5) A language with too many operators and special symbols is
   - a. good with respect to writability
   - b. bad with respect to reliability
   - c. bad with respect to readability
   - [✓ d.] all of the above

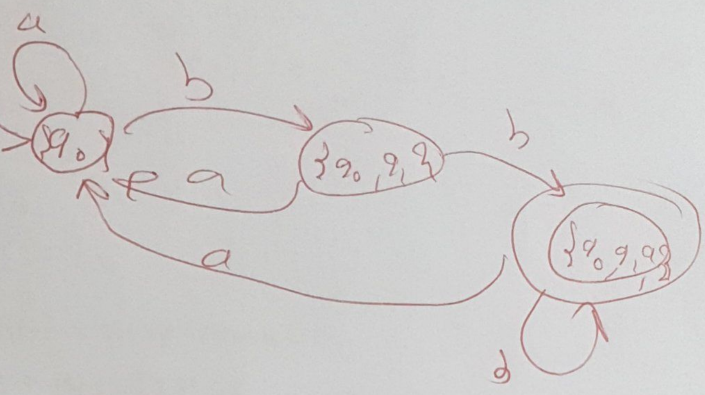
6) Java is partly compiled and partly interpreted (hybrid implementation) makes is
   - [✓ a.] Good for writing portable programs
   - b. Good for writing efficient programs
   - c. Good for reducing the cost of training
   - d. All of the above

7) Classes, inheritance and polymorphism designed in programming languages as a result of
   - a. The Von-Neumann architecture
   - [✓ b.] Programming methodologies
   - c. People preferences
   - d. None of the above

Q2) Consider the following NFA



A) Convert the following NFA into a DFA (4 marks)



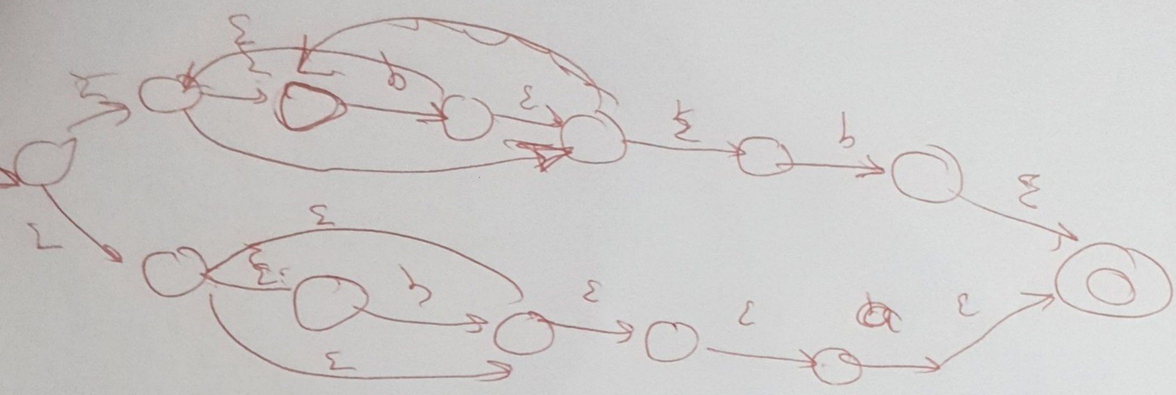
B) Represent the DFA as a table (2 marks)

	a	b
{q0}	{q0}	{q0, q1}
{q0, q1}	{q0}	{q0, q1, q2}
{q0, q1, q2}	{q0}	{q0, q1, q2}

C) Write an algorithm determines if a string belongs to the language of the DFA or not. (3 marks)

```
state = S
i = 0
while (input [i])
        state = T(state, input [i++]);
if (state = q2)
        accept
else reject
```

D) Convert the regular expression (a*b) + (b*a) into an NFA (3 marks)

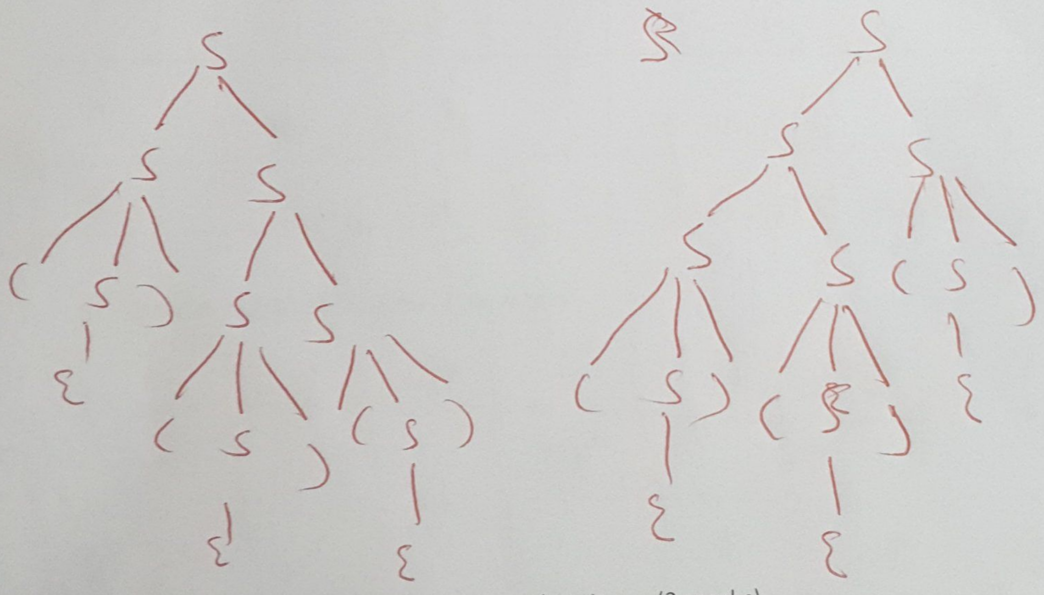


Q3) Consider the following CFG

S→ (S) | [S] | SS | ε

A) Show that the grammar is ambiguous. (3 marks)

The string ()()() has more than one parse tree



B) Rewrite the grammar to eliminate ambiguity. (3 marks)

one possible solution

S ⟶ (S)S | [S]S | ε

C) Without drawing the LL(1) parsing table, give two reasons why the given grammar is not an LL(1) grammar. (3 marks)

1. ambiguous

2. left recursion

Q4) Consider the following CFG

E → T E'
E' → + T E' | ε
T → F T'
T' → * F T' | ε
F → (E) | id

A) Find the first and follow sets for each non-terminal symbol. (10 grades)

Symbol	First set	Follow Set
E	{ (, id }	{ $,) }
E'	{ +, ε }	{ $,) }
T	{ (, id }	{ +, $,) }
T'	{ *, ε }	{ +, $,) }
F	{ (, id }	{ *, +, (, id }

B) Draw the LL(1) parsing table (4 marks)

	+	*	(	)	id	$
E			TE'		TE'	
E'	+TE'			ε		ε
T			FT'		FT'	
T'	ε	*FT'		ε	ε	
F			(E)		id	

C) Is the grammar an LL(1) grammar? Why or why not. (2 marks)

Yes it is because there are no multiply defined entries.