Programming Language Compilation
Sample Exam Questions
CSC 340
Mid-1 2nd Term 2021-2022
KSU

Student Name: [redacted]
Student Number: [redacted]
Student Section: 9 - 10

18

Q1 (10 grades) Put a circle around the most appropriate answer for each of the following

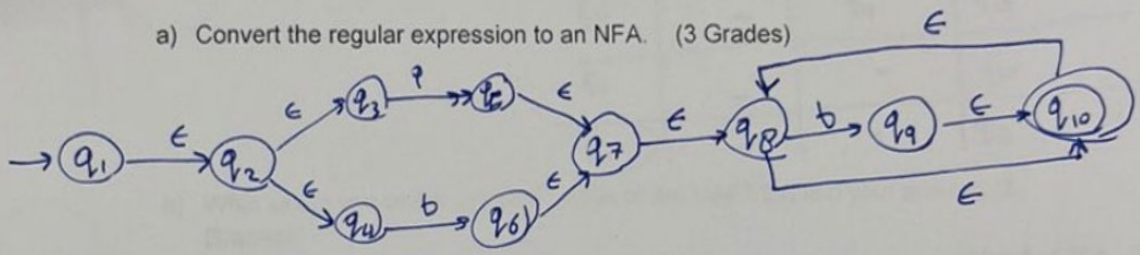
1. Orthogonality makes a programming language good with respect to
   a. Readability
   b. Writability
   c. Reliability
   d. All of the above
2. The ability to use pointers in C++, makes the language
   a. Good with respect to writability but bad with respect to reliability
   b. Good with respect to reliability but bad with respect to writability
   c. Good with respect to reliability and writability
   d. Good with respect to readability but bad with respect to reliability
3. The fact that Java checks the range of array indices, makes the language
   a. Good with respect to writability but bad with respect to reliability
   b. Good with respect to the cost of execution but bad with respect to reliability
   c. Good with respect to reliability but bad with respect to the cost of execution
   d. None of the above
4. We usually prefer to stop adding features to an existing language, and begin a new one to
   a. Reduce the cost of re-training programmers
   b. make use of the most recent development in the field of programming languages
   c. Reduce the cost of writing a program
   d. Increase reliability
5. Java uses a combination of compilation and interpretation to
   a. achieve portability and reduce the cost of translation
   b. reduce the cost of training programmers
   c. improve reliability and writability
   d. all of the above
6. Which of the following features make a programming language suitable for scientific applications
   a. Floating-point representations
   b. Data persistency
   c. Low level control on hardware
   d. All of the above

7. Which of the following features makes a programming language good with respect to reliability
   a. Exception handling
   b. Type checking
   c. Limited aliasing
   d. All of the above

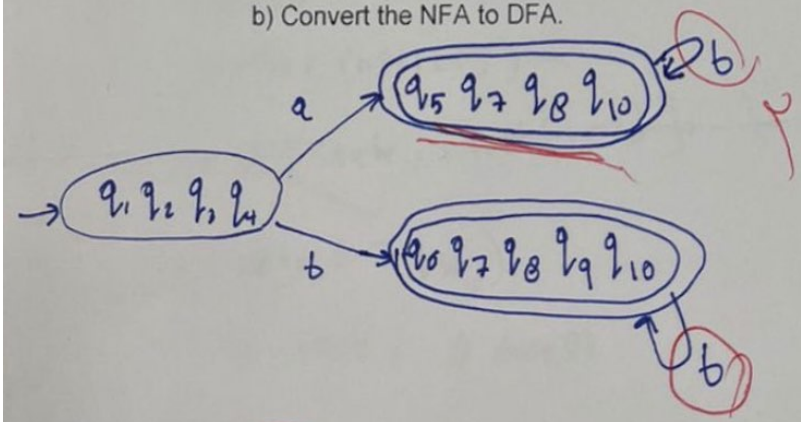
8. Java is better than C with respect to reliability because
   a. It checks the array index range
   b. It contains exception handling
   c. Does not have pointers
   d. All of the above

Q2. (14 grades) Consider the language L given by the regular expression $(a+b)b^*$ over the alphabet $\{a,b\}$.

a) Convert the regular expression to an NFA. (3 Grades)



b) Convert the NFA to DFA. (4 Grades)



Q3) (8 grades) Describe each of the following languages over the alphabet S={a,b} using regular expressions and Context-Free Grammars if possible. If it is not possible to do either of them explain why.

a) A string must contains an odd number of a

1. Regular expressions (2 Grades)

• b^*a

2. Context-Free Grammar (2 Grades)

$S \rightarrow aSb \mid bSa \mid b$

b) The number of a's in the string is twice the b's.

1. Regular expressions (2 Grades)

$(baa)^*$

2. Context-Free Grammar (2 Grades)

$S \rightarrow aSb \mid bSa \mid a$

DFA

b) Represent the above DFA and NFA using a tables. (2 Grades)

State	a	b
$q_1 q_2 q_3 q_0$	$q_5 q_7 q_8 q_{10}$	$q_6 q_7 q_8 q_9 q_{10}$
$q_5 q_7 q_8 q_{10}$	-	$q_5 q_7 q_8 q_{10}$
$q_6 q_7 q_8 q_9 q_{10}$	-	$q_6 q_7 q_8 q_9 q_{10}$

NFA

State	a	b	ϵ
q_1	-	-	q_2
q_2	-	-	$\{q_3, q_4\}$
q_3	q_5	-	-
q_4	-	q_6	-
q_5	-	-	q_7
q_6	-	-	q_7
q_7	-	-	q_8
q_8	-	q_9	q_{10}
q_9	-	-	q_{10}
q_{10}	-	-	q_8

2

b) What would you prefer to use the DFA or the NAF? Explain your answer. (2 Grades)

It's trade-off, I will use DFA if I want fast execution, without care about cost, and memory.

and use NFA if I Perfer to save memory. (in most sinar but not always.

c) Write a complete pseudo-code that uses DFA or NAF (depending on your answer above) to determine if a string belongs to the language L. The code must display a suitable accept or reject message. So if your answer above was DFA then you must write a code that uses the DFA table to but if your answer was an NFA then the code must use the NFA table. (3 Grades)

```
// Assume I used NFA.
i=0
State=0
while(input[i]) {
state= A[state, input[i++]] }
if(State == q10)
return true; // Accept
else
return false; // Reject
```

King Saud University
College of Computer and Information Sciences
Computer Science Department

Course Code	CSC 340		
Course Title	Programming Language and Compilation		
Section No.			
Semester	S2 – 1442/1443		
Exam	Midterm 2		
Date	7 April 2022	Duration	1 Hour
Student Name	[redacted]		
Student ID	[redacted]		

Course Learning Outcomes

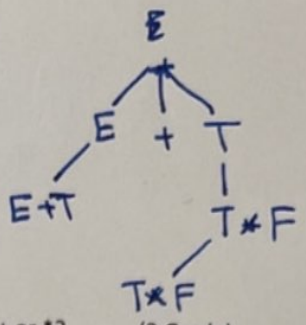
		Relevant question	Full mark	Student mark
CLO 1				9
CLO2				9
CLO 3				8
CLO 4				
CLO 5				
CLO 6				

Feedback/Comments:

24/32

Q1) Consider the following CFG

E → E+T | T
T → T*F | F
F → (E) | id

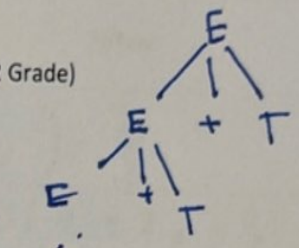


A) Which operation has a higher precedence + or *? (2 Grade)

* has higher Precedence,

B) What is the associativity of the + operation? (2 Grade)

left Associativity,



C) What is the associativity of the * operation? (2 Grades)

left

D) Rewrite the grammar to reverse the 1) precedence and 2) associativity of all operations. (3 Grades)

① ②

① Precedence

E → E*T | T
T → T+F | F
F → (E) | id

E → T* E | T
T → F+T | F
F → (E) | id

Final Step

E) Rewrite the grammar given in the question to make it suitable for recursive descent parsing. (3 Grades)

E → XT
X → E+ | ∈
T → ZF
Z → T* | ∈
F → (E) | id

Q2) Consider the following grammar

S → RS'
S' → - R S' | ε
R → FR'
R' → /FR' | ε
F → (S) | id

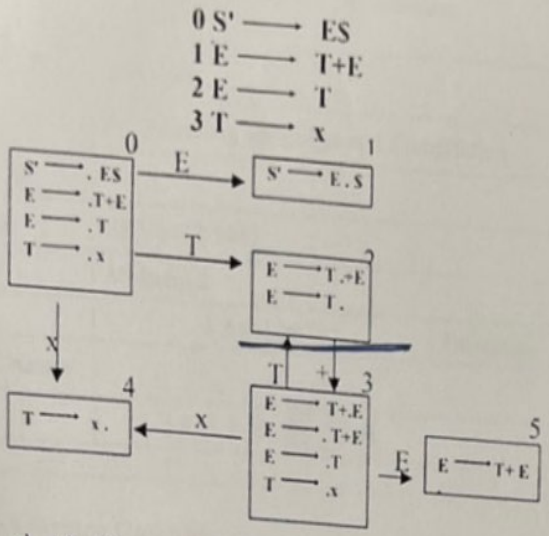
a) Find the first and follow sets for all symbols in the following table (7 Grades)

Symbol	First	Follow
S	First(R) → {(, id}	{$,)}
S'	{-, ∈}	Follow(S) → {$,)}
R	First(F) → {(, id}	{-, $,)}
R'	{/, ∈}	Follow(R) → {-, $,)}
F	(, id	First(R') → {/, -, $,)} Follow(R)
-	{-}	{-, $,)}
Id	{id}	{(, id}

b) Write down the LL(1) parsing table. (4 Grades)

	-	/	(	)	id	$
S			RS'		RS'	
S'	-RS'			∈		∈
R			FR'		FR'	
R'	∈	/FR'		∈		∈
F			(S)		id	
-	-RS'					
Id	id					

Q3) Consider the following CFG and the DFA that recognizes the viable prefixes for SLR parsing



a) Identify any conflict for an SLR parser? List the state number and the type of conflict it contains if any. (2 Grades)

State 2

Shif - Recuce conflict

b) Is the grammar an SLR? Explain your answer (3 Grades)

Yes,

It's Un Ambigous,

b) Show the main steps for parsing the expression x+x$. Use a suitable table to illustrate your answer. (4 Grades)

	Step	Action
$\|x+x$	1	Shift
x|+x$	4	Recuce
T|+x$	1	Shift
T+|x$		Shift
T+x|$		Reduce
T+T|$	2	Reduse
T+E|$		Reduse
E|$		Reduce
		Shift

E$ | 0 | Reduce

S | 0 | Accept