28
38

Programming Language Compilation
Midterm-1
CSC 340
KSU (2nd term 2017-2018)

Student Name: [redacted] **Student Number:** [redacted]

Q1) a) Which programming language is better with respect to reliability Java or C++? Explain, why? (4 marks)

Java, because it check the index~~ing~~ of the array if it out of bound or not.

4

and C++ have Pointers and they are not reliable.

b) Which programming language is better with respect to the cost of execuation Java or C++? Give at least 2 reasons. (4 marks)

Answer: C++

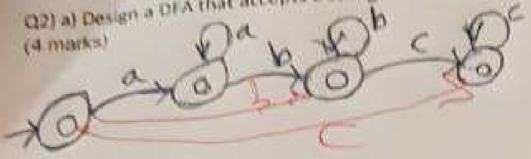
Reasons:

1. C++ does not check for Proper indexing, it allows ~~to~~ the user to go out of bound in the array

4

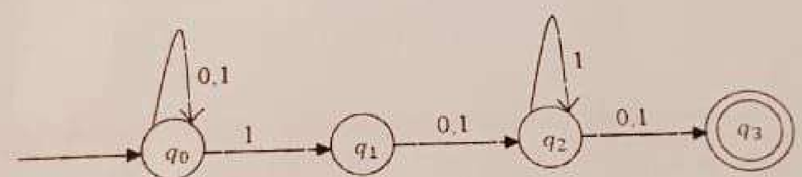
2. ~~C++~~ the compiler of C++ translate to machine languge ~~immedi~~ diretly, but in Java it transtate to intermadite languge then to machine languge.

Q2) a) Design a DFA that accepts a string if it belongs to the language $\{a^p b^q c^r \mid p, q, r \geq 0\}$ (4 marks)



2

b) Consider the finite automaton in the following figure.



1. What is the set of reachable states for the input string 0011? (2 marks)

$\{q_0, q_1, q_2\}$

2

2. Is the string 0011 acceptable? Why? (2 marks)

2

NO, because the reachable state with this input ~~are~~. are $\{q_0, q_1, q_2\}$ and non of them ~~at~~ belongs to F

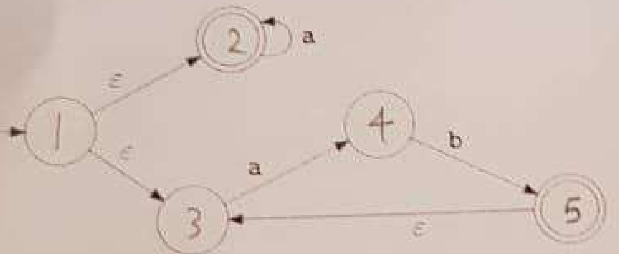
$\{q_0, q_1, q_2\} \notin F$ "set of final states"

d) Represent the above finite automaton using a table. (2 marks)

	0	1
→q_0	q_0	$\{q_0, q_1\}$
q_1	q_2	q_2
q_2	q_3	$\{q_3, q_2\}$
*q_3	—	—

2

Q3) Consider the following NFA

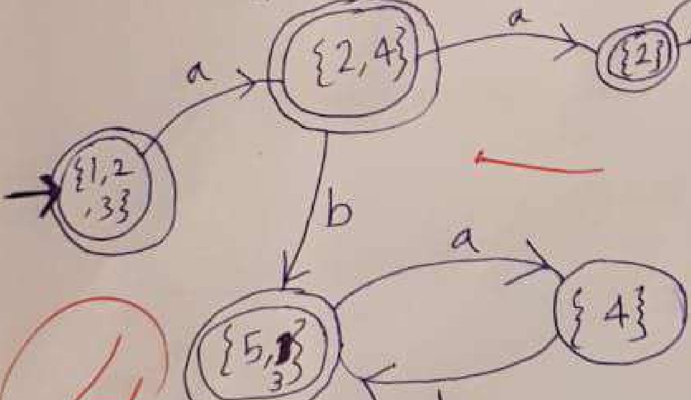


a) Describe the language recognized by the NFA using a regular expression. (2 marks)

$a^* + (ab)^+$

2

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



4

ε-close{1} = {1,2,3}

NOTE

	a	b
→*{1,2,3}	{2,4}	—
*{2,4}	{2}	{5,3}
*{2}	{2}	—
*{5,3}	{4}	—
{4}	—	{5,3}

c) Represent the DFA using a Table (1 mark)

	a	b
→*{1,2,3}	{2,4}	—
*{2,4}	{2}	{5,3}
*{2}	{2}	—
*{5,3}	{4}	—
{4}	—	{5,3}

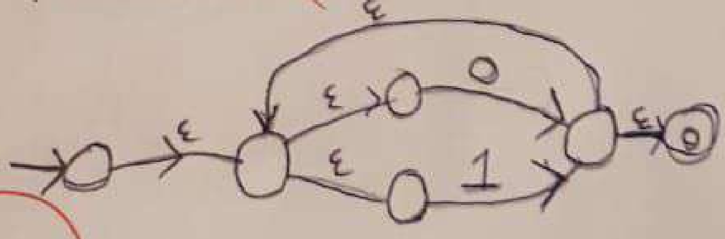
1

d) Write an algorithm that takes a string over the alphabet and displays "accept" if the string is acceptable by the DFA, or "reject" if it is not. (3 marks)

```
inPut= inPut string
state =0  // start state
while (inPut){
state = table [state , inPut++]
}
if (state ∈ F)  // F is set of final states
    return ("accePt")
else
    return ("reject")
```

2

e) Convert the regular expression $(0+1)^*$ over the alphabet {0,1} into an NFA (3 marks)



3