Programming Language Compilation
Midterm (1st term 2020-2021)
CSC 340
KSU

Student Number: **Student Name:**

Q1) a) Compare between Java and C++ with respect to reliability. (2 marks)

- Java performs array index checking; C++ does not
- Java uses references which are much cleaner than pointers as used in C++
- Java is a strongly typed language while C++ is weakly typed

=> Java is more reliable

b) Discuss how pointers in C++ affects

1) writability (1 marks)

Good for writability as they allow us to use vars, array, objects using their addresses in the memory

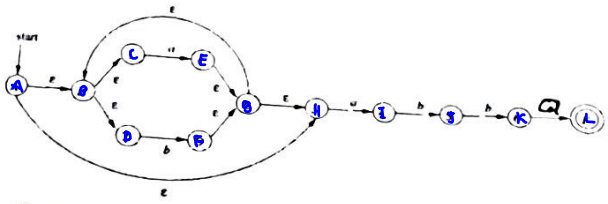
2) reliability (1 marks)

Bad in terms of reliability as mistakes can happen, especially if we use pointer arithmentics

c) Why were modular languages developed? (2 marks)

- To support huge applications
- divide and conquer approach for development
- data abstraction

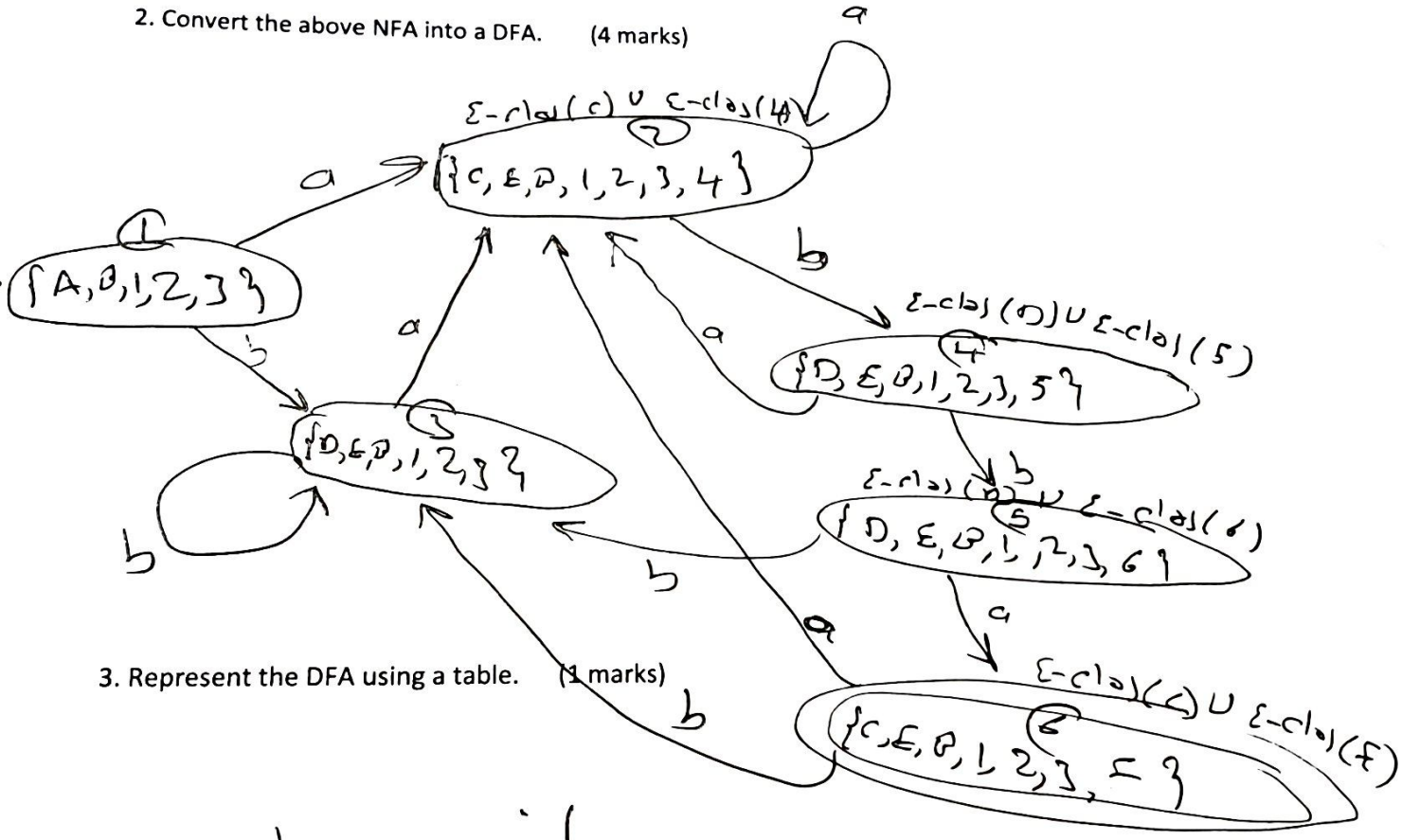
Q2) Consider the following NFA



1. Describe the language accepted by this NFA using a regular expression. (1 marks)

$$(a+b)^* abba$$

2. Convert the above NFA into a DFA. (4 marks)



3. Represent the DFA using a table. (1 marks)

State	a	b
1	2	3
2	2	4
3	3	2
4	2	5
5	6	3
6	2	3

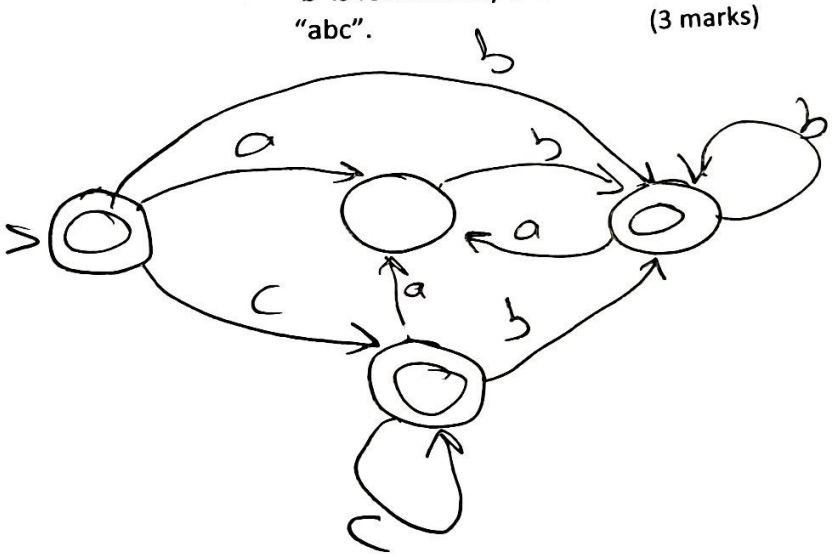
4. Write an algorithm that uses the above table to decide if a string is acceptable by the DFA or not. (2 marks)

```
i = 0
state = 0
while (input[i])
    state = T[state, input[i++]]

if state = 6
    then accept
else
    reject
```

Q3) a) Consider the alphabet the alphabet S= {a, b, c}.

A) Design a DFA that accepts a string in which every 'a' is followed by a 'b', but no 'b' is followed by a 'c'. For example, the DFA should accept "cabbab" but not "abc". (3 marks)



B) Write a regular expression that describes the language accepted by the above DFA. (2 marks)

$c^* b^* (ab)^*$

Q4) a) Consider the following CFG

E → E - int | E / int | int

a) Which operation has the highest precedence? (1 marks)

- the leftmost operation

b) What is the association of the – operation? (1 marks)

- Left to right

c) What is the association of the / operation? (1 marks)

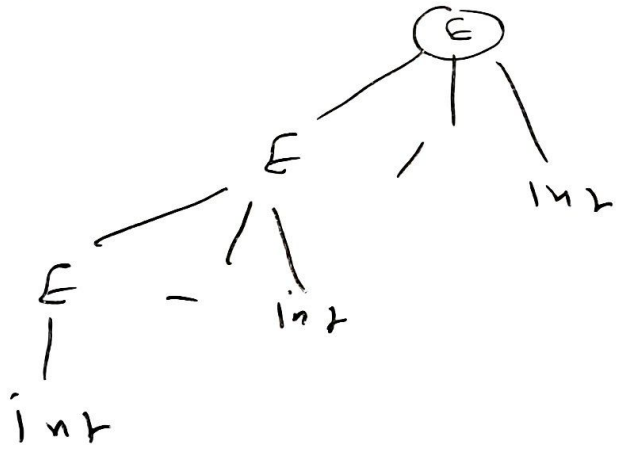
- Left to right

d) Show that the string ***int-int/int*** belongs to the language of the above CFG (valid). (1 marks)

- either

E ⇒ E/int ⇒ E-int/int ⇒ int-int/int

or



e) Write a CFG that describes all mathematical expressions that can be formed using only * and +. Makes sure that * has higher precedence over + and the associativity of both operations is left to right. (3 marks)

E → E' + int | E'

E' → E' * int | E | int