



## SEG2106 B –Software Construction

### Midterm Exam

Professor: Ayman El-Sawah

March 1<sup>st</sup> 2019, duration 80 Minutes

### Identification

Student name: \_\_\_\_\_

Student number: \_\_\_\_\_ Signature: \_\_\_\_\_

### Instructions

1. No calculators, electronic devices, or other aids permitted;
2. Write your answers in the space provided;
3. You may not hand in additional pages
4. Poor hand writing may affect grades

### Marking Scheme

	Q(s)	Max	Mark
P1	1-10	10	
	1	5	
	2	5	
	3	5	
	4	5	
	5	10	
	6	5	
Total		45/40	

## Part 1 – Multiple Choice Questions (10 Marks)

For the multiple choice questions, **Circle the single best answer**. There is only one best answer for each question. Questions

1. In Agile process development \_\_\_\_\_ is responsible for sprint meeting
  - a. Product owner
  - b. Scrum team
  - c. **Scrum master**
  - d. All the above
2. In Agile process development who prioritizes product backlog?
  - a. **Product owner**
  - b. Scrum team
  - c. Scrum master
  - d. All the above
3. Which of these statements are truly acceptable?
  - a. A precondition is an assertion guaranteed to be true when the activity or operation finishes
  - b. A post condition is an assertion guaranteed to be true when the activity or operation begins
  - c. **Trigger is an event which cause a use case to begin**
  - d. None of the mentioned
4. What is true in context to use case extensions?
  - a. Once the basic flow is defined, the extensions can be specified
  - b. The alternatives are called extension as they extend the activity flow in a different direction from branch point
  - c. **All of the mentioned**
  - d. None of the mentioned
5. Select the correct statement about the Specification and Description Language (SDL):
  - a. Signals are considered synchronous events exchanged between instances of processes
  - b. You can create at most one process instance from each process definition
  - c. **Channels are used to connect blocks to the environment**
  - d. A parent process cannot send a signal to an “offspring” process it has created

6. Select the correct meaning of the SDL constructs

- |                     |   |
|---------------------|---|
| A) Receive Message  | 1  |
| B) Send Message     | 2  |
| C) Decision         | 3  |
| D) Procedure Call   | 4  |
| E) Create a Process | 5  |
| F) Save             | 6  |

- a. A1, B2, C3, D4, E5, F6
- b. A1, B2, C6, D3, E4, F3
- c. A2, B1, C5, D4, E3, F6
- d. **A2, B1, C5, D3, E4, F6**
- e. A1, B2, C6, D4, E3, F5

7. Which of the following statements about actions in a UML2 activity diagram are true?

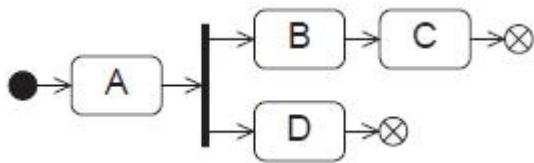
S1	Actions can manipulate objects and their values	T
S2	Actions cannot be aborted	F
S3	Actions are atomic	T
S4	An action consists of several activities	F

- a. TTTT
- b. TFTF
- c. FTFT
- d. TFTT

8. Petri nets \_\_\_\_\_

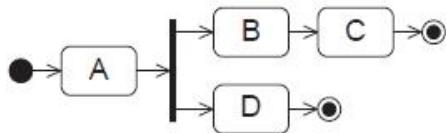
- a. Are unbounded when there is at least one transition to trigger at all times
- b. **Are structural models used to analyse the design of distributed systems**
- c. Consist of three types of components: places, transitions, and states
- d. Are considered deadlocked if most transitions can no longer fire

9. You are given the following activity diagram. Which of the following action sequences are possible during one execution of the activity diagram?



S1	A→B→C	F
S2	A→B→D→C	T
S3	A→D→C	F
S4	A→D→B→C	T
S5	A→B→C→D	T
S6	A→D	F
S7	A→B→D	F

- a. FTFTFTF  
 b. FTFTFFF  
 c. TTTTTTT  
 d. TFTFTFT
10. Which of the following action sequences are possible during one execution of the following activity diagram?



S1	A→B→D→C	F
S2	A→B→C	T
S3	A→D	T
S4	A→B→C→D	F
S5	A→B→D	T

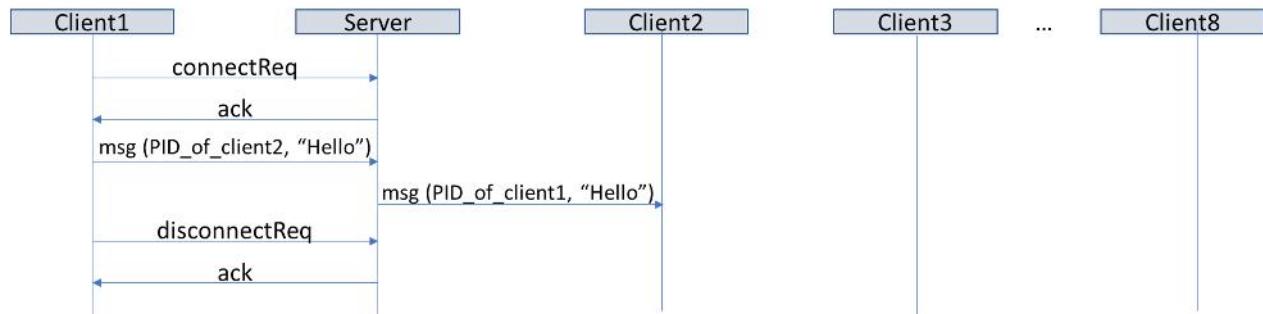
- a. TFTTF  
 b. FTTFT  
 c. FTFTT  
 d. TTTTT

## Part 2 – (Development Questions)

1. Consider a chat server that forwards messages between connected clients. Hence, a client can 1) connect to the server, 2) send a message to another client, and 3) disconnect. The Server can support the simultaneous connection of 8 clients. (5 Marks)

Figure 1 shows a Message Sequence Chart (MSC) depicting the interaction between the clients and server. The MSC shows the following exchange of information:

1. Client1 sends a connection request (**connectReq**) to the server
2. Server acknowledges that the connection request is accepted (**ack**)
3. Client1 sends a chat message (**msg**) to Client2 by specifying the PID of Client2 and the text for the message it wishes to send ("Hello" in the example depicted in Figure 1). Note that we are assuming in this example that Client2 is already connected (the Client2 connection sequence of messages is not shown in Figure 1).
4. Server forwards the message to Client2 while specifying the PID of the sender (Client1). Specifying the PID of the sender (i.e. Client1) allows Client2 to respond to Client1 if it wishes to do so.



**Figure 1: MSC for the interaction between the Clients and Server**

Figure 2 shows the state machine for the Server process. Below are few notes about the Server state machine:

- The integer variable **connectedClients** keeps track of how many clients are connected at a time. The behavior of the Server process does not allow more than 8 clients to connect simultaneously to the Server.
- **connectionsMap** is a hash map where the key is a Process ID (**PID**) and the value is a **BOOLEAN**. This hash map allows the Server to keep track of connected clients. When a client connects to the Server, a hash map entry is added where the key is the **PID** of the client that just connected and the value is **TRUE**. When that client disconnects, the value is switched to **FALSE**.

- When the maximum number of allowable connections is reached and a client attempts to connect to the server, then a not acknowledged (**nack**) signal is sent to the client to inform them that a connection is not possible at the moment.
- If the client is currently connected, and attempts to connect again, then a **nack** signal is returned to the client.

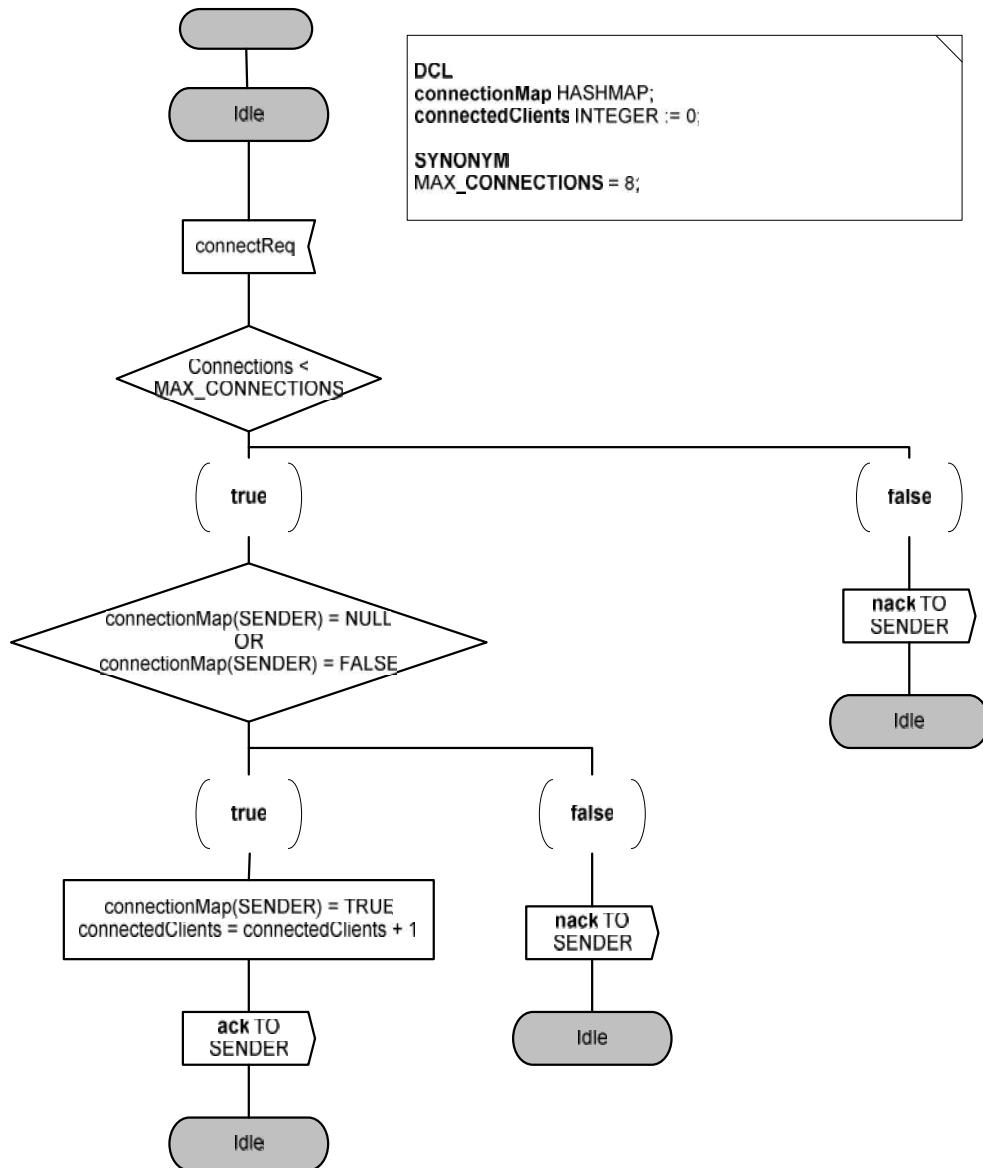
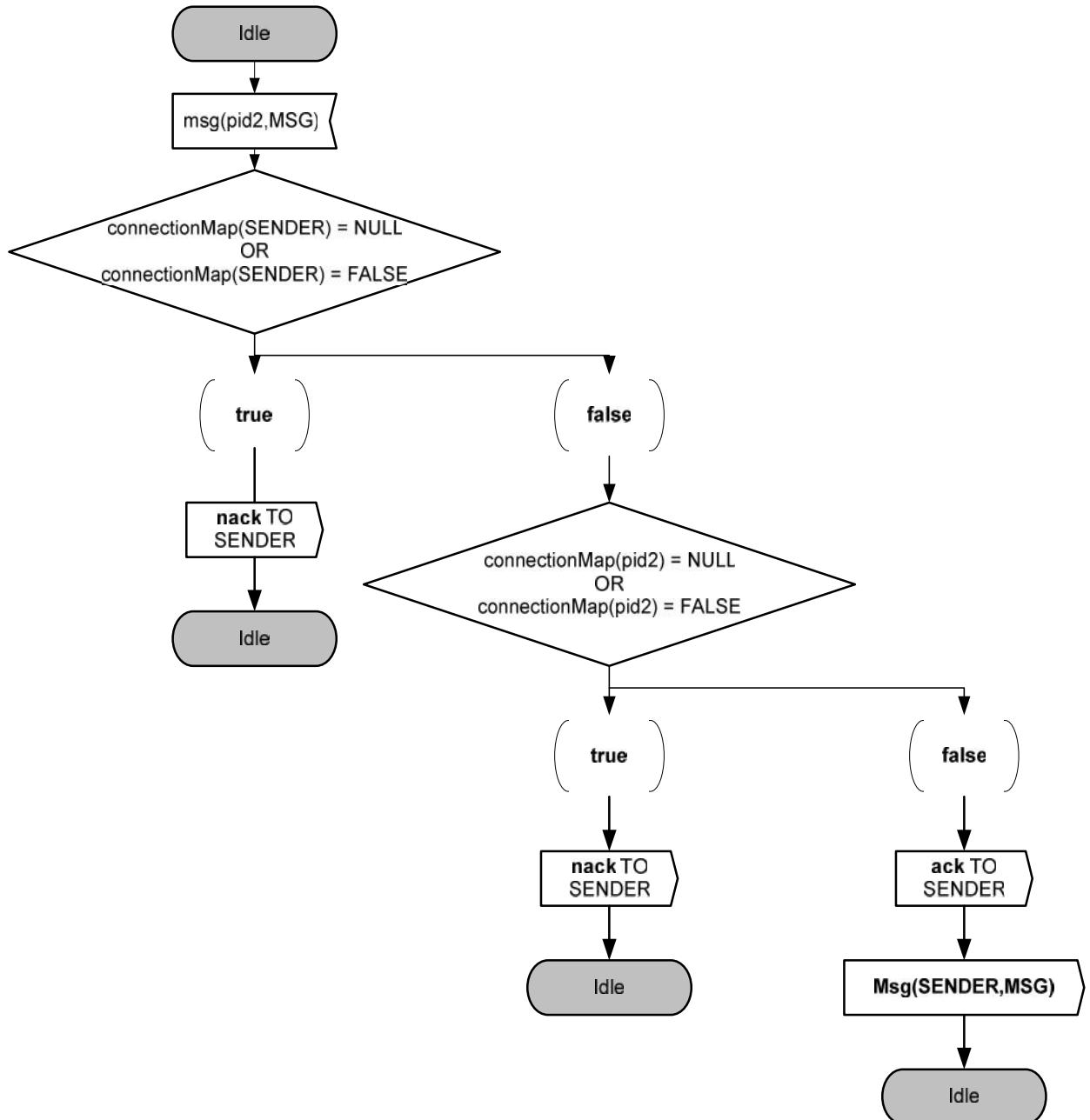


Figure 2: Server state machine

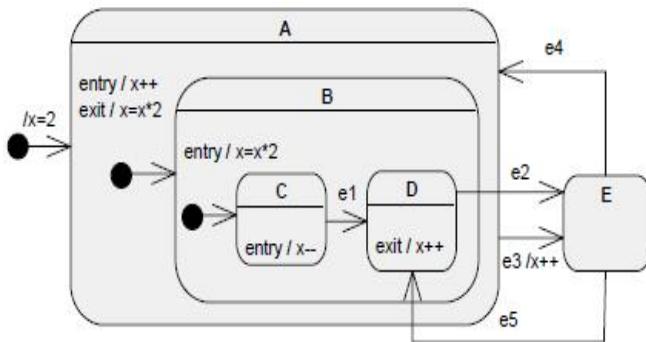
Your Task is to add the logic for forwarding the message received from one client to another client as described the scenario shown in Figure 1. Show only your extension/addition, i.e. no need to replicate existing state machine.

Hint: (Extend the Server's state machine to respond to client's msg). Ensure that an unconnected client cannot send a message to other clients. If an unconnected client attempts to send a message to another client, the Server should return a **nack** signal to that client. Also ensure that a client cannot send to unconnected clients, if the intended recipient is not connected, the Server should return a **nack** signal to that client.

(Show your answer here)



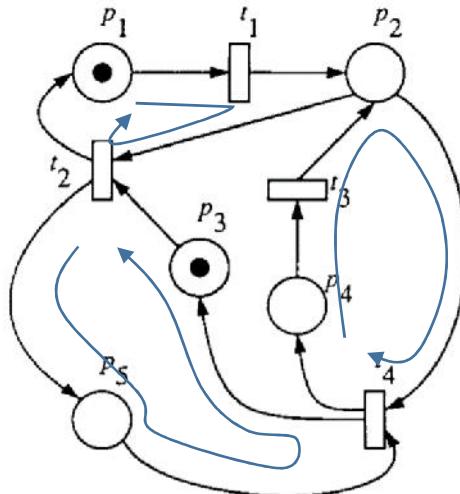
2. You are given the following state machine diagram. What is the value of  $x$  after the occurrence of the event chain **e1 e2 e4 e2 e1**? (5 Marks)



Event	state	value
	A	2
entry	A	3
entry	B	6
entry	C	5
e1	D	5
exit	D	6
exit	A	12
e2	E	12
e4	A	12
entry	B	26
entry	C	25
e2	C	25
e1	D	25

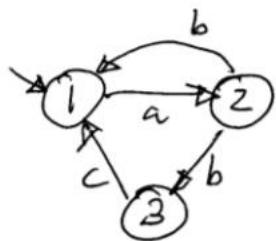
**Answer:  $x = 25$**

3. Examine the Petri Net shown below and answer the following questions. (5 Marks)



- Is the Petri Net bounded? **Yes**
- Is the Petri Net safe? **Yes**
- Is the Petri Net Live? **Yes**
- Is the petri Net Reversible? **No**
- Compare between  $p_1$  &  $p_3$  in terms of frequency of token visits.  $\frac{P_1}{P_3} = \frac{2}{1}$

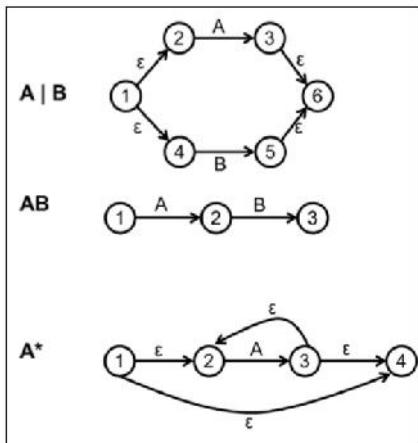
4. The figure below shows a non-deterministic finite automata (NFA) with alphabet  $\Sigma = \{a,b,c\}$  (5 Marks)



- Convert NFA above to deterministic finite automata (DFA). Show your derivation.
- Write a regular expression for the language generated by the automata. Assume that state '3' is the only accepting state.

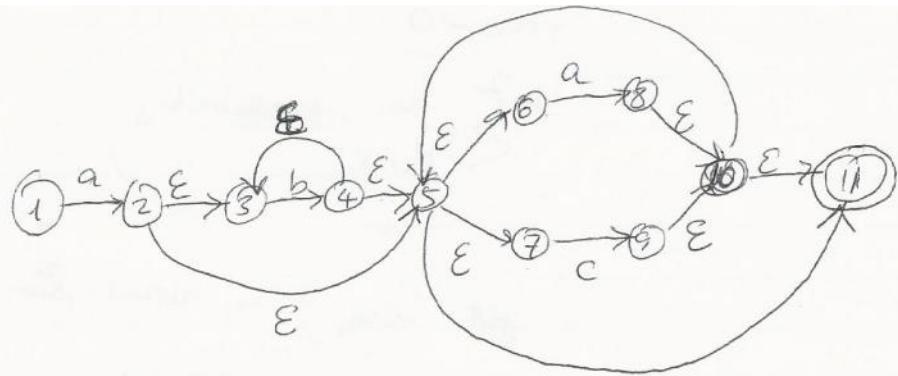
$\begin{aligned} CL(\{1\}) &= \{1\} = A \\ CL(T(A,a)) &= CL(\{2\}) = \{2\} = B \\ CL(T(A,b)) &= \{\} = \emptyset \\ CL(T(A,c)) &= \{\} = \emptyset \\ CL(T(B,a)) &= \{\} = \emptyset \\ CL(T(B,b)) &= \{1,3\} = C \\ CL(T(B,c)) &= \{\} = \emptyset \\ CL(T(C,a)) &= \{2\} = B \\ CL(T(C,b)) &= \{\} = \emptyset \\ CL(T(C,c)) &= \{1\} = A \end{aligned}$	<pre> graph LR     S(( )) --&gt; A((A))     A -- a --&gt; B((B))     A -- c --&gt; C(((C)))     B -- a --&gt; A     B -- b --&gt; C     C -- a --&gt; B   </pre> <p><b>ab (cab ab)*</b></p>
--	---

5. Convert the regular expression:  $ab^*(a|c)^*$  to a Non-Deterministic Finite Automata (NFA) using the basic rules given below, then find an equivalent Deterministic Finite Automata (DFA). Show the resulting state transition table for the resulting DFA. (10 Marks)



state	a	b	c	accepting
A	B			
B	C	D	E	yes
C	C		E	yes
D	C	D	E	yes
E	C		E	yes

State Transition Table



$$\bar{S} = S \cap A$$

$$T(A/a) = \bar{S} = \{2, 3, 5, 6, 7, 11\} = B$$

$$T(A/b) = T(A/c) = \emptyset$$

$$T(B/a) = \bar{S} = \{8, 10, 5, 6, 7, 11\} = C$$

$$T(B/b) = \bar{S} = \{4, 5, 6, 7, 11, 3\} = D$$

$$T(B/c) = \bar{S} = \{9, 10, 5, 6, 7\} = E$$

$$T(C/a) = \bar{S} = C$$

$$T(C/b) = \emptyset$$

$$T(C/c) = \bar{S} = E$$

$$T(D/a) = \bar{S} = C$$

$$T(D/b) = \{4\} = D$$

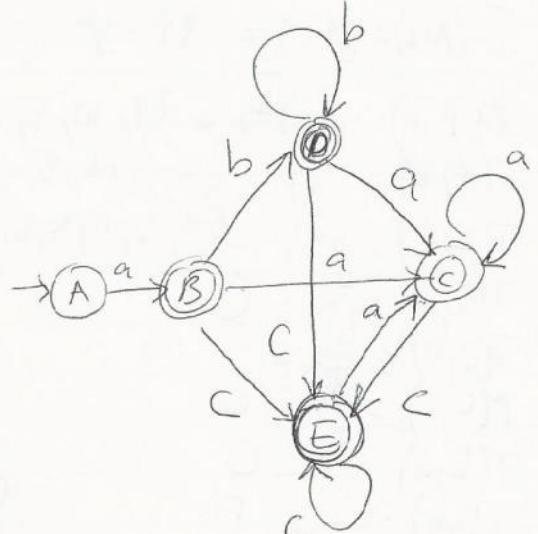
$$T(D/c) = \{9\} = E$$

$$T(E/a) = \bar{S} = C$$

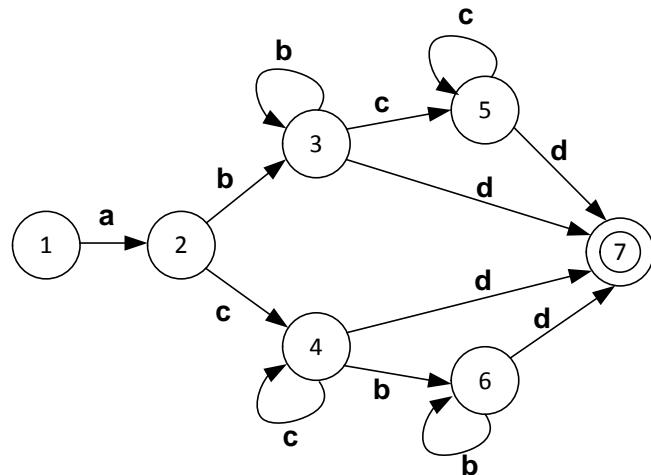
$$T(E/b) = \emptyset$$

$$T(E/c) = \bar{S} = E$$

	a	b	c	Accept
A	B			
B	C	D	E	✓
C	C		E	✓
D	C	D	E	✓
E	C		E	✓



6. Find the minimized deterministic finite automata (DFA) of the following DFA. (5 Marks)



	a	b	c	d	Accept	Step	comment
1	2					2	Consume 'a' – only one
2		3	4	7		7	Move to 4 on c
3		3	5	7		6	Move to 5 on c
4		6	4	7		5	Move to 6 on b
5			5	7		4	Does not consume b – only one
6		6		7		3	Does not consume c – only one
7					yes	1	Terminal must be separate

Already optimal!