AQA Computer Science A-Level

4.4.2 Regular languages

Past Paper Questions

Additional Spec Qs Paper 1

0 4 **Figure 3** contains a list of different sets.

Figure 3

S1 = {a, b}
S2 = {a, b, c}
S3 = {0, 1, 2}
S4 = {a, ab}
S5 = {a, b, c}
S6 = {c}

0 4 . 1 What is a set?

[2 marks]

0 4 . 2 State the name of a set listed in **Figure 3** that has the same cardinality as set S2.

[1 mark]

0 4 . 3 State the name of a set listed in **Figure 3** that is a subset of set S2 but not a proper subset of S2.

[1 mark]

0 4 . 4 Describe how set S6 could be created using the difference set operation together with two of the other sets listed in **Figure 3**.

[1 mark]

0 4 . 5 What is the Cartesian product of sets S1 and S6?

[1 mark]

A regular expression is a way of describing a set.

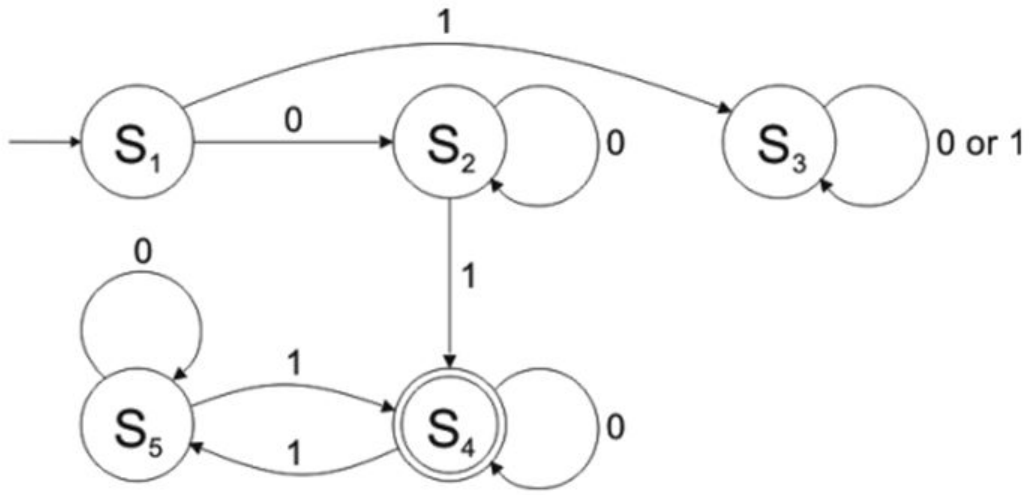
0 4 . 6 Write down **two** different regular expressions that describe the set S4.

[2 marks]

June 2011 Comp 3

4 **Figure 2** shows a Finite State Automaton (FSA). The FSA has input alphabet {0, 1} and five states, S_1, S_2, S_3, S_4 and S_5.

Figure 2



4 (a) Complete the transition table below for the FSA in **Figure 2**.

Current State	S_1	S_1	S_2	S_2	S_3	S_3	S_4	S_4	S_5	S_5
Input Symbol	0	1	0	1	0	1				
Next State	S_2	S_3	S_2	S_4	S_3	S_3				

(1 mark)

4 (b) The state S_4 is a special state. This is indicated by the double circle in the diagram. What does the double circle signify?

.....

.....

(1 mark)

4 (c) Write **Yes** or **No** in each row of the table below to indicate whether or not each of the four input strings would be accepted by the FSA in **Figure 2**.

Input String	String Accepted? (Yes/No)
101	
000	
010001101	
0100011011	

(2 marks)

4 (d) Describe the language (set of strings) that the FSA will accept.

.....

.....

.....

(2 marks)

9 Regular expressions can be used to search for strings.

9 (a) For each of the following regular expressions, describe the set of strings that they would find.

9 (a) (i) `a+b`

.....

.....

(1 mark)

9 (a) (ii) `a?b`

.....

.....

(1 mark)

9 (a) (iii) `(ab)*`

.....

.....

(1 mark)

9 (b) Write regular expressions that match:

9 (b) (i) either `Clare` or `Claire`.

.....

(1 mark)

9 (b) (ii) any non-empty string that:

- starts with `10`
- has zero or more occurrences of any combination of `0` or `1` in the middle
- ends with `01`

Example strings that the expression should match are `1001`, `100010101`, `101111010101001`.

.....

(2 marks)

June 2012 Comp 3

The system uses four different voltage levels so that two data bits can be transmitted with each signal change.

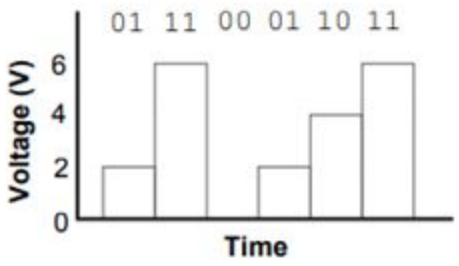
Table 3 shows the signal levels (in volts) that the system uses for particular binary patterns.

Table 3

Binary pattern	Signal level (volts)
00	0
01	2
10	4
11	6

Using this system, the binary pattern 011100011011 would be transmitted as the voltage sequence 2,6,0,2,4,6 as shown in **Figure 4**:

Figure 4

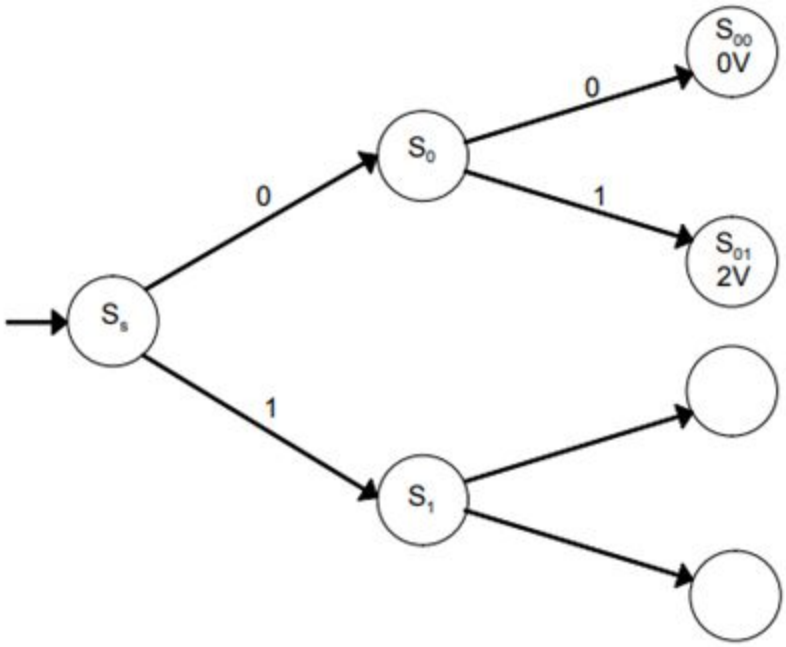


4 (d) A Moore machine is a type of finite state machine that produces output. The transitions are labelled with the inputs and each state is labelled with a name and the output that it produces; if a particular state has no output then it is labelled with just a name.

Figure 5 shows an incomplete diagram of a Moore machine that will convert a two-bit binary code into the signal level (in volts) that is transmitted to represent it, as listed in **Table 3**.

Complete **Figure 5**. Label all of the transitions and the states that are currently unlabelled. The machine should work for the four binary patterns 00, 01, 10 and 11.

Figure 5



(4 marks)

12 Regular expressions can be used to search for strings. For example, `de(f|g)*h+` matches any string that starts with `de` and is followed by zero or more instances of either `f` or `g` followed by one or more instances of `h`.

Write regular expressions that will match:

12 (a) any string that starts with a letter `a`, ends with a letter `c` and has one or more occurrences of the letter `b` in the middle of it, ie the expression should match the strings `abc`, `abbc`, `abbbc` and so on.

.....

(1 mark)

12 (b) any string that starts with either a `0` or a `1`, followed by zero or more occurrences of the digit `1` ie the expression should match the strings `0`, `1`, `01`, `11`, `011` and so on.

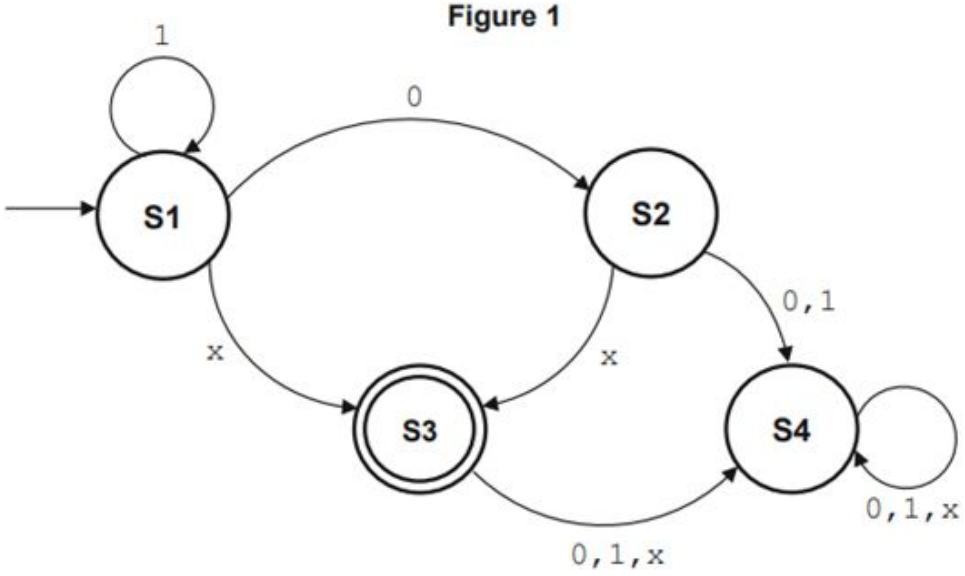
.....

(1 mark)

June 2016 AS Paper 1

0 2 The finite state machine (FSM) represented as a state transition diagram in **Figure 1** recognises a language with an alphabet of `0`, `1` and `x`.

Figure 1



Input strings of `0x` and `1x` are accepted by this FSM.

0 2 . 1 In **Table 1** indicate whether each input string is accepted or not accepted by the FSM in **Figure 1**.

If an input string is accepted write YES.
If an input string is **not** accepted write NO.

Complete **Table 1** by filling in the unshaded cells.

Copy the contents of all the unshaded cells in **Table 1** into your Electronic Answer Document.

Table 1

Input string	Accepted by FSM?
111011x	
1110x	
111001x	

[2 marks]

0 2 . 2 In words, describe the language (set of strings) that are accepted by the FSM in **Figure 1**.

[3 marks]

June 2017 AS Paper 1

0 1 A hotel provides a safety deposit box in guest rooms. The safety deposit box has a keypad with twelve buttons, as shown in **Figure 1**.

Figure 1

1	2	3
4	5	6
7	8	9
C	0	E

The safety deposit box operates as follows:

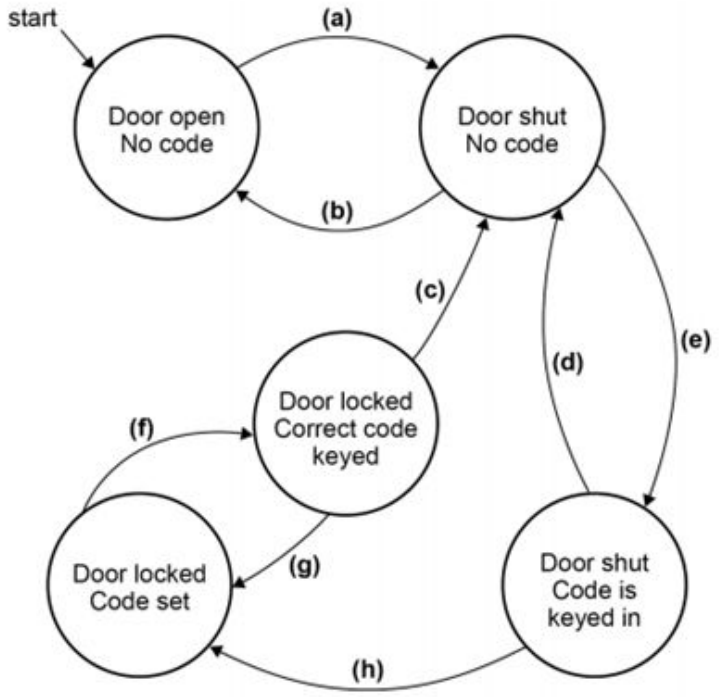
The buttons with digits 0 to 9 enable the guest to set their own code. Button C cancels any digits entered. Button E is the Enter key.

- To close the safety deposit box:
  - push the door shut
  - key in a new 4-digit code (guest's choice)
  - press the Enter key (this sets the code and locks the door).
- To open the safety deposit box:
  - key in the correct 4-digit code (previously chosen by the guest)
  - press the Enter key (this also deletes the stored code)
  - pull open the door.
- Pressing the keypad has no effect, except when keying in a code.

Figure 2 shows a partially complete state transition diagram that represents the operation of the safety deposit box. The events are labelled **(a)** to **(h)**.

Note the state transition diagram does not show what happens if an incorrect code is keyed in.

Figure 2



0 1 . 1 In **Table 1** indicate which label(s), **(a)** to **(h),** represent(s) which event. Two labels have to be assigned to some of the events.

Complete **Table 1** by filling in the unshaded cells with the correct labels from **Figure 2**. A label **must** only be used once.

Table 1

Event	Label(s): (a) to (h)
Correct code keyed	
Door pulled open	
Door pushed shut	
New code keyed	
Press C	
Press E	

Copy the contents of all the unshaded cells in **Table 1** into your Electronic Answer Document.

[4 marks]

June 2017 Paper 1

0 2 Postcodes are used to aid the sorting of mail and help to ensure that mail being sent arrives at the correct destination as quickly as possible.

The format of a UK postcode (ignoring any spaces) is shown in **Figure 2**.

Figure 2

- 1 or 2 letters
- followed by:
  - 1 numeric digit or
  - 2 numeric digits or
  - 1 numeric digit then 1 letter
- followed by 1 numeric digit
- followed by 2 letters

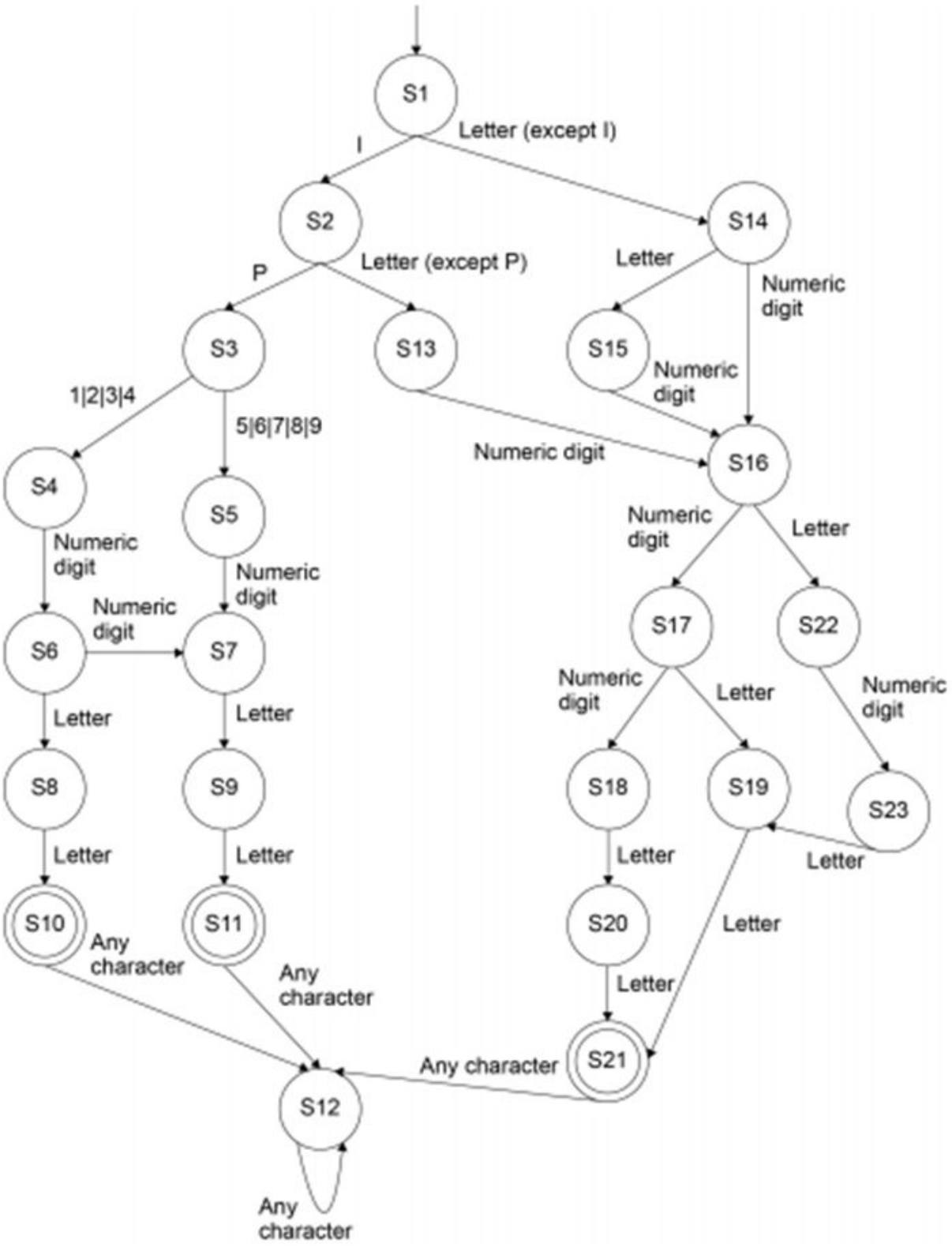
When a post box is emptied in the town of Ipswich the mail in the post box is taken to a central sorting office. Each item is looked at and placed in one of three vans depending upon the postcode written on the envelope.

Postcodes that begin with IP1, IP2, IP3 or IP4 followed by one numeric digit and two letters, eg IP2 8QY, are for mail being sent to an address in the town of Ipswich and go in Van A. Other postcodes that begin with IP, eg IP5 3QW, are for areas not in the town but near to Ipswich and go in Van B. Postcodes that start with anything other than IP, eg CO3 5FN, are not for the Ipswich area and go in Van C. IP postcodes do not use the full range of formats available for UK postcodes.

A finite state machine (FSM) could be used to sort mail using postcodes. **Figure 3** shows a state transition diagram for an FSM used at the Ipswich sorting office.

In **Figure 3**, if a transition is not defined from a state for a particular input symbol then the FSM will stop processing the input and it will be rejected.

Figure 3


S1
Letter (except I)
I
S2
S14
Letter (except P)
P
Letter
Numeric digit
S3
S13
S15
Numeric digit
1|2|3|4
5|6|7|8|9
Numeric digit
S16
S4
S5
Numeric digit
Letter
Numeric digit
Numeric digit
Numeric digit
S17
S22
S6
S7
Numeric digit
Letter
Numeric digit
Letter
Letter
S8
S9
S18
S19
S23
Letter
Letter
Letter
Letter
S10
S11
S20
Letter
Any character
Any character
Letter
Letter
Any character
S21
S12
Any character

0 2 . 1 If the FSM in **Figure 3** reaches state S12 what does it mean?

[1 mark]

0 2 . 2 If the FSM in **Figure 3** finishes at state S11 what does it mean?

[1 mark]

0 2 . 3 Assuming that the FSM in **Figure 3** can be used to recognise any valid IP postcode, state **one** format used for UK postcodes that IP postcodes do **not** use.

[1 mark]

Figure 2 is repeated below.

Figure 2

- 1 or 2 letters
- followed by:
  - 1 numeric digit or
  - 2 numeric digits or
  - 1 numeric digit then 1 letter
- followed by 1 numeric digit
- followed by 2 letters

0 2 . 4 The language recognised by an FSM can also be represented by a regular expression. When writing regular expressions `\d` is used to represent any numeric digit and `\a` is used to represent any alphabetic character.

For example, the regular expression `\d \d \a \d` describes the language of all strings that contain two numeric digits followed by one letter and then one numeric digit.

Write a regular expression that represents a valid UK postcode as described in **Figure 2**. In your answer you should only use the `|` metacharacter once.

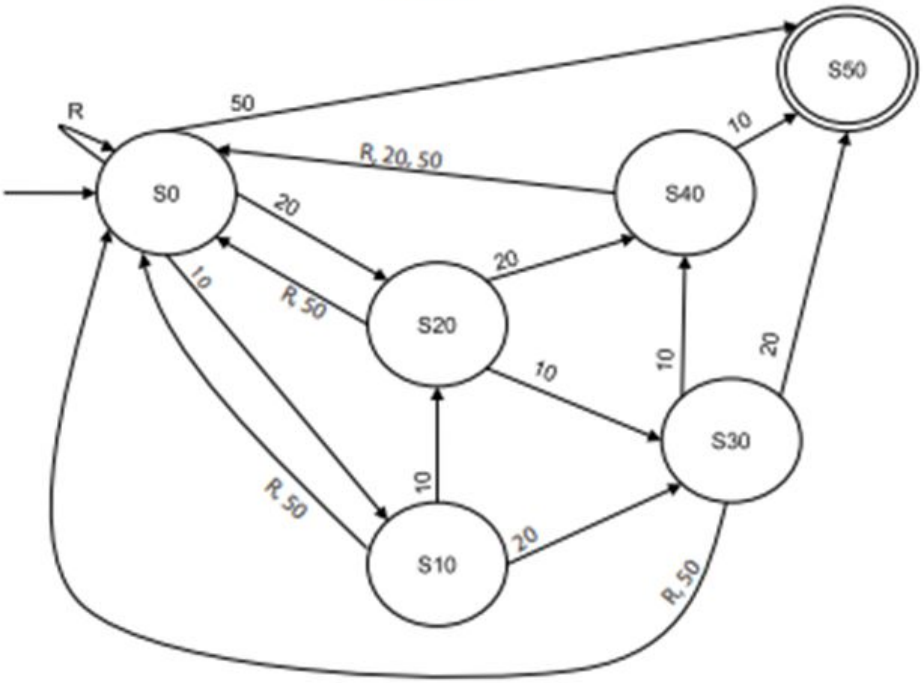
[4 marks]

June 2012 Comp 1

Figure 2 shows the state transition diagram of a finite state machine (FSM) used to control a vending machine.

The vending machine dispenses a drink when a customer has inserted exactly 50 pence. A transaction is cancelled and coins returned to the customer if more than 50 pence is inserted or the reject button (R) is pressed. The vending machine accepts 10, 20 and 50 pence coins. Only one type of drink is available.
The only acceptable inputs for the FSM are 10, 20, 50 and R.

Figure 2



0 9 An FSM can be represented as a state transition diagram or as a state transition table. **Table 2** is an incomplete state transition table for part of **Figure 2**.

Complete the missing sections of the four rows of **Table 2**.

*Copy the cells in **Table 2** that contain your answer into the Electronic Answer Document.*

Table 2

Original state	Input	New state
S0	10	S10
S0		
S0		
S0		

(3 marks)

There are different ways that a customer can provide **exactly three** inputs that will result in the vending machine dispensing a drink. Three possible permutations are "20, 10, 20", "10, R, 50" and "10, 50, 50".

1 0 List **four** other possible permutations of **exactly three** inputs that will be accepted by the FSM shown in **Figure 2**. *(4 marks)*

June 2013 Comp 1

Figure 1 shows a state transition diagram for a finite state machine (FSM).

Table 1 shows the outputs produced by the finite state machine in **Figure 1** for some possible input strings. Some of the outputs are missing from **Table 1**. Input strings are processed starting with the right-most bit.

Figure 1

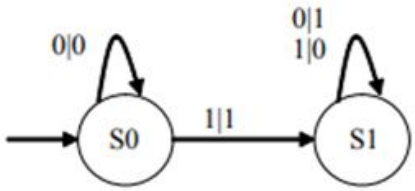


Table 1

Input string	Output string
00010011	11101101
00010010	**(a)**
00010100	11101100
00010101	**(b)**

0 6 What output string should be in position **(a)** in the table? *(1 mark)*

0 7 What output string should be in position **(b)** in the table? *(1 mark)*

0 8 What is the purpose of the finite state machine shown in **Figure 1**? *(1 mark)*

A finite state machine can be represented as a state transition diagram or as a state transition table. **Table 2** is an incomplete state transition table for **Figure 1**.

0 9 *Complete the **unshaded** cells in the table in the Electronic Answer Document that correspond to the unshaded cells in **Table 2** below.*

Table 2

Input	Original state	Output	New state
0	S0	0	S0
1		1	S1
0	S1		S1

(3 marks)

7 A particular Turing machine has states S_B, S_0, S_1, S_R and S_T. S_B is the start state and S_T is the stop state. The machine stores data on a single tape which is infinitely long in one direction. The machine's alphabet is 0, 1, #, x, y and □ where □ is the symbol used to indicate a blank cell on the tape.

The transition rules for this Turing machine can be expressed as a transition function δ. Rules are written in the form:

$$\delta \text{ (Current State, Input Symbol)} = \text{(Next State, Output Symbol, Movement)}$$

So, for example, the rule:

$$\delta (S_B, 1) = (S_1, y, \rightarrow)$$

means:

IF the machine is currently in state S_B AND the input symbol read from the tape is 1

THEN the machine should change to state S_1, write a y to the tape and move the read/write head one cell to the right

The machine's transition function, δ, is defined by:

$$\begin{aligned}
\delta (S_B, 0) &= (S_0, x, \rightarrow) \\
\delta (S_B, 1) &= (S_1, y, \rightarrow) \\
\delta (S_B, \#) &= (S_T, \#, \rightarrow)
\end{aligned}$$

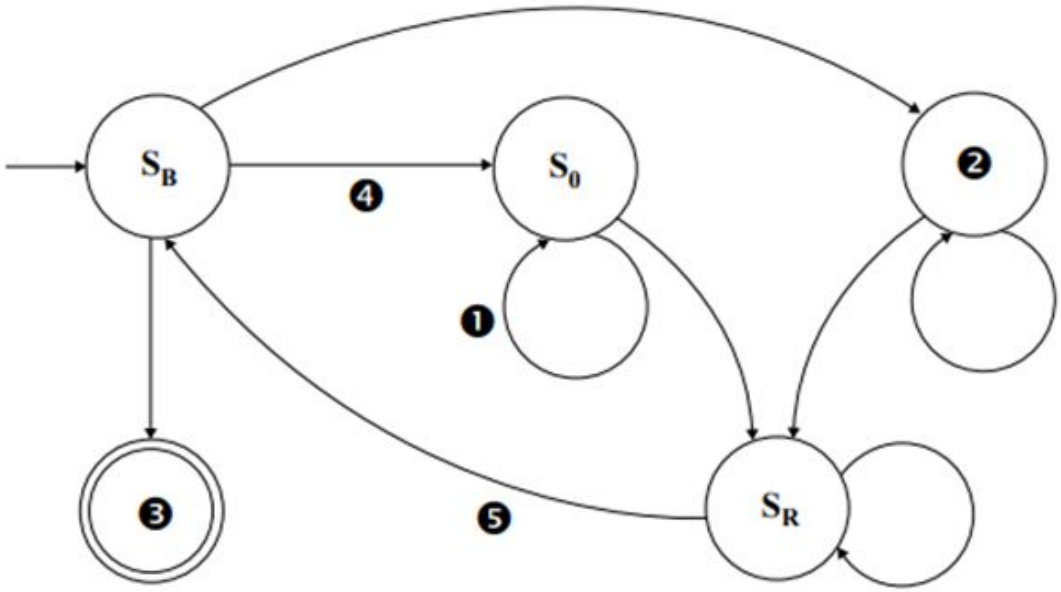
$$\begin{aligned}
\delta (S_0, 0) &= (S_0, 0, \rightarrow) \\
\delta (S_0, 1) &= (S_0, 1, \rightarrow) \\
\delta (S_0, \#) &= (S_0, \#, \rightarrow) \\
\delta (S_0, \square) &= (S_R, 0, \leftarrow)
\end{aligned}$$

$$\begin{aligned}
\delta (S_1, 0) &= (S_1, 0, \rightarrow) \\
\delta (S_1, 1) &= (S_1, 1, \rightarrow) \\
\delta (S_1, \#) &= (S_1, \#, \rightarrow) \\
\delta (S_1, \square) &= (S_R, 1, \leftarrow)
\end{aligned}$$

$$\begin{aligned}
\delta (S_R, 0) &= (S_R, 0, \leftarrow) \\
\delta (S_R, 1) &= (S_R, 1, \leftarrow) \\
\delta (S_R, \#) &= (S_R, \#, \leftarrow) \\
\delta (S_R, x) &= (S_B, 0, \rightarrow) \\
\delta (S_R, y) &= (S_B, 1, \rightarrow)
\end{aligned}$$

Figure 9 shows an unlabelled finite state transition diagram for this machine. Some of the state transition arrows represent more than one of the machine's transition rules. For example, the arrow labeled ❶ represents the three rules: $\delta (S_0, 0) = (S_0, 0, \rightarrow)$, $\delta (S_0, 1) = (S_0, 1, \rightarrow)$ and $\delta (S_0, \#) = (S_0, \#, \rightarrow)$.

Figure 9



7 (a) (i) Which states are represented by the labels ❷ and ❸ in **Figure 9**?

❷ ..... ❸ .....

(1 mark)

7 (a) (ii) Which of the machine's transition rule(s) is/are represented by the arrow labelled ❹ in **Figure 9**?

.....

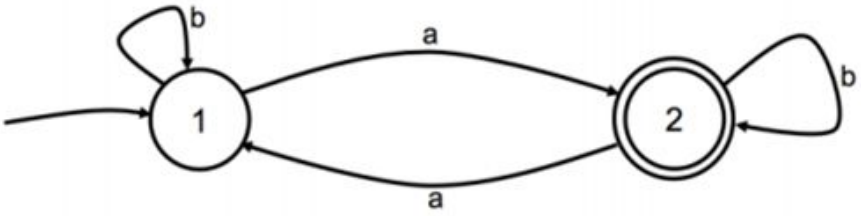
.....

(1 mark)

Specimen AS Paper 1

The finite state machine (FSM) shown in **Figure 2** recognises a language with an alphabet of `a` and `b`.

Figure 2



Input strings of `a` and `aabba` would be accepted by this FSM.

0 1 . 6 In **Table 1** indicate whether each input string would be accepted or not accepted by the FSM in **Figure 2**.

If an input string would be accepted write YES.
If an input string would **not** be accepted write NO.

Copy your answer in **Table 1** into the Electronic Answer Document.

Table 1

Input string	Accepted by FSM?
aaab	
abbab	
bbbbba	

[2 marks]

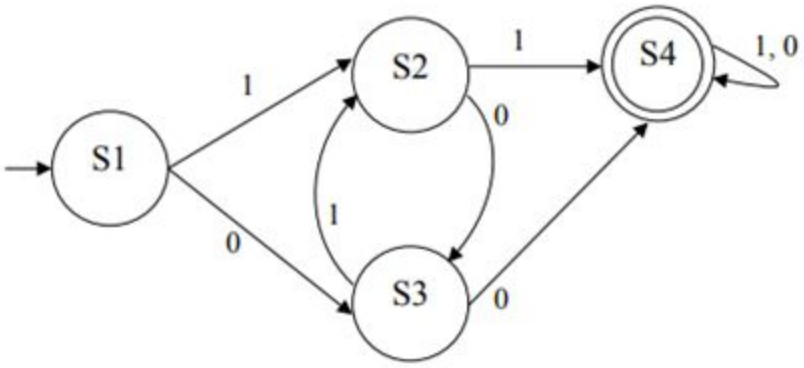
0 1 . 7 In words, describe the language (set of strings) that would be accepted by this FSM shown in **Figure 2**.

[2 marks]

Specimen Paper 1

0 2 A finite state machine (FSM) can be used to define a language: a string is allowed in a language if it is accepted by the FSM that represents the rules of the language.
Figure 1 shows the state transition diagram for an FSM.

Figure 1



An FSM can be represented as a state transition diagram or as a state transition table.
Table 1 is an incomplete state transition table for **Figure 1**.

0 2 . 1 Complete **Table 1** and copy the table into the Electronic Answer Document.

Table 1

Original state	Input	New state
S3		
S3		

[1 mark]

Any language that can be defined using an FSM can also be defined using a regular expression.

The FSM in **Figure 1** defines the language that allows all strings containing at least, either two consecutive 1s or two consecutive 0s.

The strings 0110, 00 and 01011 are all accepted by the FSM and so are valid strings in the language.

The strings 1010 and 01 are not accepted by the FSM and so are not valid strings in the language.

0 2 . 2 Write a regular expression that is equivalent to the FSM shown in **Figure 1**.

[3 marks]