AQA Computer Science AS Level

3.4.2 Finite state machines (FSMs)

Past Paper Mark Schemes

June 2011 Comp 3

4 (a)

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	***0***	***1***	***0***	***1***
Next State	S_2	S_3	S_2	S_4	S_3	S_3	***S_4***	***S_5***	***S_5***	***S_4***

1 mark for all four bolded columns correct
A the two columns for S_4 either way round and similar for S_5 **1**

4 (b)

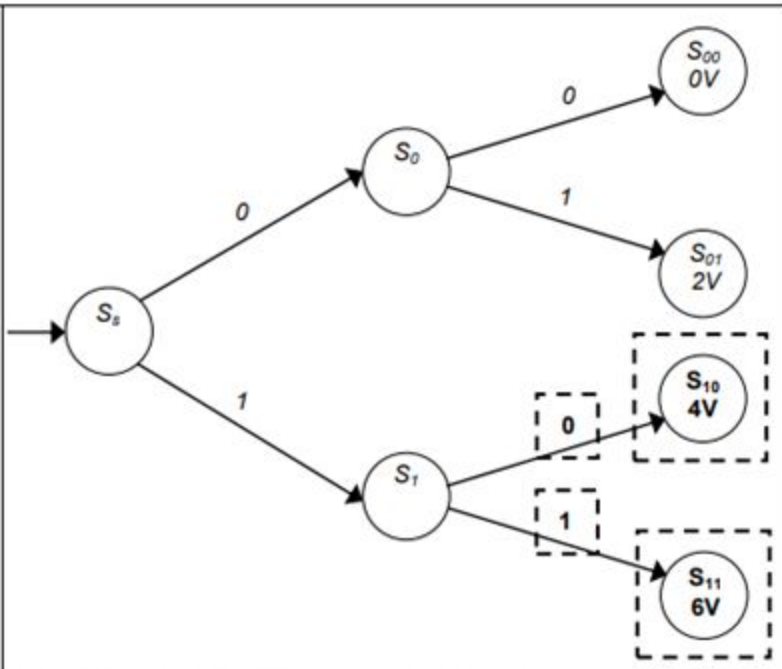
Accept/Accepting/Accepted (state) // Input (string) is accepted
A if the FSA finishes in this state output is Yes
R Stop state **1**

4 (c)

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

1 mark for any two correct answers
2 marks for all four answers correct **2**

June 2012 Comp 3

4	(d)		 **1 mark** for labelling a transition arrow with 0 **1 mark** for labelling a transition arrow with 1 **1 mark** for labelling a state with the value 4V and a unique state name **1 mark** for labelling a state with the value 6V and a unique state name **MAX 2 if the states and transition arrow labels do not correspond** Note that: • The state names do not have to match those given here. • The voltage values can be followed by a V, the word Volts or nothing. • The zero and one on the transition arrows to the right of S_1 can be either way around e.g. 1 above 0 is okay.	4

June 2016 AS Paper 1

<table>
<tr><td>02</td><td>1</td><td>All marks AO2 (apply)

<table><tr><th>Input string</th><th>Accepted by FSM?</th></tr><tr><td>111011x</td><td>NO</td></tr><tr><td>1110x</td><td>YES</td></tr><tr><td>111001x</td><td>NO</td></tr></table>
Mark as follows:
1 mark: one row correct
2 marks: all rows correct</td><td>2</td></tr>
<tr><td>02</td><td>2</td><td>All marks AO2 (apply)

Strings that start with zero or more 1s; A. starts with any number of 1s as BOD

which may or may not be followed by a 0; A. there can be at most one 0 in the string

and end with an x; A. 'end' being by implication

NOTE: 'ending with either x or 0x' is worth two marks

NOTE: MAX 2 if answer is not fully correct</td><td>3</td></tr>
</table>

June 2017 AS Paper 1

<table>
<tr><th>Qu</th><th></th><th></th><th>Marks</th></tr>
<tr><td>01</td><td>1</td><td>All marks AO2 (apply)

<table>
<tr><th>Event</th><th>Label(s)</th></tr>
<tr><td>Correct code keyed</td><td>F</td></tr>
<tr><td>Door pulled open</td><td>B</td></tr>
<tr><td>Door pushed shut</td><td>A</td></tr>
<tr><td>New code keyed</td><td>E</td></tr>
<tr><td>Press C</td><td>d, g (I. order)</td></tr>
<tr><td>Press E</td><td>h, c (I. order)</td></tr>
</table>
1 mark per two correct labels (round down).

I. case

Note: each label must only be used once (if given more than once, reject all occurrences).</td><td>4</td></tr>
</table>

June 2017 Paper 1

02	1	**Mark is for AO2 (analyse)** Input string is a (valid) postcode followed by additional characters // the input string is not a valid (UK) postcode // the mail will not be put in any of the three vans; **NE.** the input string is not a valid IP postcode **A.** Postcode has additional characters at the end **A.** Postcode is too long	1
02	2	**Mark is for AO2 (analyse)** (The string represents) an IP postcode that is not for a location in the town of Ipswich // (The string represents) an IP postcode that is for a location near Ipswich // (The string represents) a postcode for a letter that needs to go in Van B; **NE.** valid postcode	1

02	3	**Mark is for AO2 (analyse)**
		
		(IP / two letters) followed by number, letter, (number, letter, letter) //
		(IP / two letters) followed by number between 5 and 9, number, (number, letter, letter) //
		IP followed by 0;
		
		A. postcodes that only have one letter at the start
			1

June 2012 Comp 1

09	<table><tr><th>Original State</th><th>Input</th><th>New State</th></tr><tr><td>S0</td><td>10</td><td>S10</td></tr><tr><td>S0</td><td>20</td><td>S20</td></tr><tr><td>S0</td><td>50</td><td>S50</td></tr><tr><td>S0</td><td>R</td><td>S0</td></tr></table> ; ; ; *Note: order of completed rows not important*	**3**

10	20, 20, 10; R, R, 50;	

	10, 20, 20; 20, 50, 50; 20, R, 50;	**MAX 4**

June 2013 Comp 1

06	11101110; **R.** 01110111	**1**
07	11101011; **DPT A.** 11010111	**1**
08	Get the two's complement (of a positive binary value) // Converts a positive binary value into its negative equivalent; **A.** It inverts all bits after the first 1 is received;	**1**
09	<table><tr><th>Input</th><th>Original State</th><th>Output</th><th>New State</th></tr><tr><td>0</td><td>S0</td><td>0</td><td>S0</td></tr><tr><td>1</td><td>S0</td><td>1</td><td>S1</td></tr><tr><td>0</td><td>S1</td><td>1</td><td>S1</td></tr><tr><td>1</td><td>S1</td><td>0</td><td>S1</td></tr></table> **Mark as follows:** S0 as original state for 2nd row; 1 as output for 3rd row; Final row correct;	**3**

Specimen AS Paper 1

01	**6**	**All marks AO2 (apply)** **1 mark:** Two rows of table completed correctly; **OR** **2 marks:** All three rows of table completed correctly; **A.** Alternative indicators for YES and NO	2

Input string	Accepted by FSM?
aaab	YES
abbab	NO
bbbbba	YES

01	**7**	**All marks AO2 (apply)** **1 mark:** a string containing zero or more (**A.** 'any number of') b characters; **1 mark:** and an odd amount of a characters; **N.E.** all strings containing an odd number of characters	2

Specimen Paper 1

02	1	**Mark is for AO1 (understanding)** **1 mark:** Table completed as above **I.** order of rows	1

Original state	Input	New state
S3	0	S4
S3	1	S2