MA4016 Solutions: Problem Sheet 10

1. (a) Reading from left to right, the output is 010110.
   (b) See elsewhere.

2. (a) See elsewhere.
   (b) By reference to the state diagram, it can be seen that the input string is either 0000 or 0011
   (c) $A = \{0\}^+ \cup \{00\}\{1\}^+$.
   (d) $A = \{0000\}\{0\}^+ \cup \{00\}\{1\}^+$.

3. (a) See elsewhere.
   (b) Reading from left to right, the outputs are

$$(a)\ 011 \quad (b)\ 0101 \quad (c)\ 00011$$

   (c) M converts input of $abc..xyz$ to output of $0ab..xy$, i.e. inserts 0 in front of input string (and deletes last symbol).

4. Each machine outputs 1 when the required string is recognised, otherwise 0. The machines are as follows:

Reading inputs from left to right, a machine that recognises 0110 is

	ν		ω	
	0	1	0	1
s_0	s_1	s_0	0	0
s_1	s_1	s_2	0	0
s_2	s_1	s_3	0	0
s_3	s_1	s_0	1	0

where s_0 is the starting state, s_1 remembers 0, s_2 remembers 10 and s_3 remembers 110.

A machine that recognises 1010 is

	ν		ω	
	0	1	0	1
s_0	s_0	s_1	0	0
s_1	s_2	s_1	0	0
s_2	s_0	s_3	0	0
s_3	s_2	s_1	0	1

where s_0 is the starting state, s_1 remembers 1, s_2 remembers 10 and s_3 remembers 101.

5. The evolving partitions and minimal machines are:

(a)
$P_1 = \{\{s_0, s_5\}, \{s_1, s_4\}, \{s_2\}, \{s_3\}\}$
$P_2 = \{\{s_0\}, \{s_5\}, \{s_1, s_4\}, \{s_2\}, \{s_3\}\}$
$P_3 = P_2$
Replace s_4 by s_1.

	ν		ω	
	0	1	0	1
s_0	s_5	s_2	0	0
s_1	s_1	s_3	0	1
s_2	s_5	s_1	1	1
s_3	s_3	s_2	1	0
s_5	s_3	s_5	0	0

(b)
$P_1 = \{\{s_1, s_2, s_3, s_4, s_5, s_7\}, \{s_6\}\}$
$P_2 = \{\{s_1, s_5\}, \{s_2, s_3, s_4, s_7\}, \{s_6\}\}$
$P_3 = \{\{s_1, s_5\}, \{s_2, s_7\}, \{s_3, s_4\}, \{s_6\}\}$
$P_4 = \{\{s_1\}, \{s_5\}, \{s_2, s_7\}, \{s_3, s_4\}, \{s_6\}\}$
$P_5 = P_4$
Replace s_7 by s_2 and s_4 by s_3.

	ν		ω	
	0	1	0	1
s_1	s_6	s_3	0	0
s_2	s_3	s_1	0	0
s_3	s_2	s_3	0	0
s_5	s_6	s_2	0	0
s_6	s_5	s_2	1	0

6. These partial answers are given in terms of a 1-tape *Turing* machine where $\mathbf{\Sigma}$ contains one (non-blank) symbol (x). The answers are by no means unique. The full answers require a program listing or transition table.

   (a) $\times 3$ Machine: As each x in the input string is deleted, write 3 x's to the output string. The leftmost symbol in the output string may be placed two cells to the right of the rightmost symbol of the input string , for instance.

   (b) Parity Machine: Delete symbols in pairs until zero or one symbols are left. (Before each pair deletion, you must check that there are 2 symbols to delete.)

   (c) Divisibility by 4 Machine: One approach is an offshoot of the Parity Machine above : Delete symbols in fours until there are less than 4 symbols. If there are zero symbols, number is divisible by 4, If there are one to three x's, delete all but one to indicate that number is not divisible by 4.

   (d) Squaring Machine: One approach uses a Copying machine i.e. a machine which, given a input string of m symbols, produces an output of 2 m-symbol strings (separated by a blank).
   The Squaring machine can then be implemented as m applications of the Copying machine. This will require an additional copy of the input string to be used to keep count of the number of Copies produced. (Various clean up operations have to be done e.g. removing blanks, additional copies, etc.)

7. These partial answers are given in terms of a 1-tape *Turing* machine where $\mathbf{\Sigma}$ contains two (non-blank) symbols $(0, 1)$. The answers are by no means unique. The full answers require a program listing or transition table.

   (a) If possible, alternatively delete a 0 from the left end of the string, then a 1 from the right end of the string; if not possible, then accept string if only blanks left on tape, otherwise reject string. Repeat until not possible.

   (b) If possible, moving rightwards, delete a 0, then a 1; if not possible, then accept string if only blanks left on tape, otherwise reject string. Repeat until not possible.