

## Problem Sheet 2: Deterministic Automata

- \* 1. For each of the automata in Figure 1, give (i) the trace of the automaton running on input 12212 and (ii) the language recognised by the automaton.

- \* 2. Give finite automata to recognise each of the following:

- (a) The language of all strings over  $\{0, 1\}$  such that each string has even length.
- (b) The language of all strings over  $\{0, 1\}$  such that each string has a number of 1s that is a multiple of three.
- (c) The language of all strings over  $\{0, 1\}$  such that each string has even length and a number of 1s that is a multiple of three.

- \*\* 3. Construct a DFA to recognise Haskell floating point literals, e.g. 2.99, 23.09e+34, 0.12E-200, 1.4e1.

A general description is as follows. A *decimal literal* is a non-empty sequence of digits (0–9). A *floating point literal* is either:

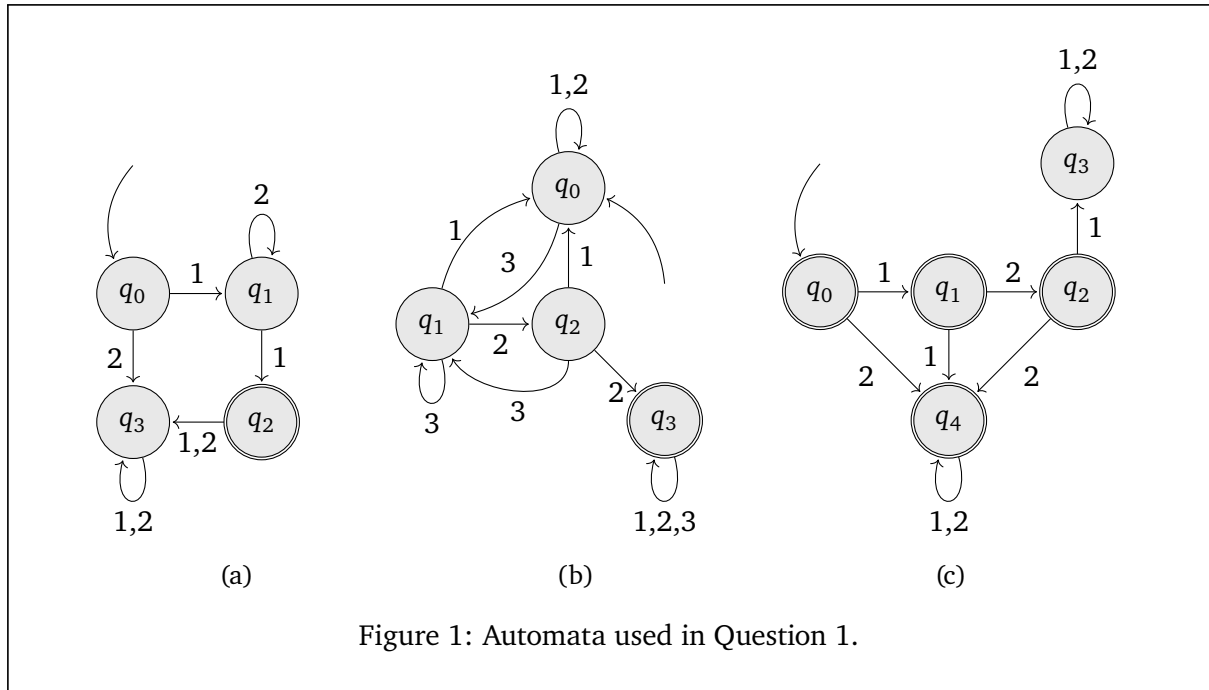
- a decimal literal followed by a decimal point followed by a decimal literal, optionally followed by an exponent
- or, a decimal literal followed by an exponent.

An exponent is either the character  $e$  or the character  $E$ ; optionally followed by the character  $+$  or the character  $-$ ; followed in all cases by a decimal literal.

Since the diagram for this automaton will be relatively complex, you may wish to adopt the following convention, which allows you to omit certain transitions:

If a state in the diagram does not have an outgoing edge for every letter of the alphabet, then it is implied that the missing edges all lead to a distinguished “junk” state, which is not an accepting state.

- \*\* 4. Suppose  $M = (Q, \Sigma, \delta, q_0, F)$  is a DFA. Construct a DFA  $\overline{M}$  with  $L(\overline{M}) = \{w \in \Sigma^* \mid w \notin L(M)\}$ . You will not be able to answer this question with a diagram.



- \*\* 5. Construct a DFA to recognise the language of strings over  $\{0, 1\}$  such that any two occurrences of 1 are separated by an even number of letters.

**Hint** how many 1s can occur in a word of this language?

- \*\*\* 6. Construct a DFA to recognise the language of strings over the alphabet  $\{a, b\}$  containing at least three occurrences of three consecutive  $b$ , overlapping permitted (e.g. the word  $bbbbbb$  should be accepted).

- \*\*\*\* 7. (Optional) Let  $\Sigma$  be the following set of binary vectors:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

We use each word  $w$  over alphabet  $\Sigma$  to encode a pair of natural numbers, written  $\llbracket w \rrbracket$ , which is defined by the following recursive function:

$$\begin{aligned} \llbracket \epsilon \rrbracket &= (0, 0) \\ \llbracket \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \cdot w \rrbracket &= (2 * m + b_1, 2 * n + b_2) \\ &\text{where } (m, n) = \llbracket w \rrbracket \end{aligned}$$

For example:

$$\begin{aligned}\llbracket \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rrbracket &= (4, 13) \\ \llbracket \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rrbracket &= (26, 3) \\ \llbracket \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rrbracket &= (1, 23)\end{aligned}$$

Construct a DFA to recognise the language of “multiplication by 3”:

$$\{w \in \Sigma^* \mid \exists m \in \mathbb{N}. \llbracket w \rrbracket = (m, 3 * m)\}$$