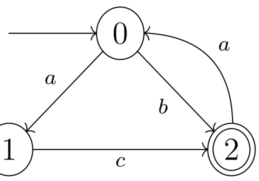
## 8 Regular Languages and Finite Automata (AMP)

(*a*) (*i*) Given any non-deterministic finite automaton $M$, describe how to construct a regular expression $r$ whose language of matching strings $L(r)$ is equal to the language $L(M)$ accepted by $M$. [5 marks]

(*ii*) Give a regular expression $r$ with $L(r) = L(M)$ when $M$ is the following non-deterministic finite automaton.




[3 marks]

(*b*) State the Pumping Lemma and explain how it is used to prove that languages are not regular. [4 marks]

(*c*) Are the following languages regular? Justify your answer in each case.

(*i*) $L_1 = \{a^k b^m c^n \mid (k = m \text{ or } m = n) \text{ and } k + m + n \geq 2\}$

(*ii*) $L_2 = \{a^k b^m c^n \mid (k = m \text{ or } m = n) \text{ and } k + m + n \leq 2\}$

(*iii*) $L_3 = \{a^k b^m c^n \mid k + m + n \geq 2\}$ [8 marks]

## 8 Regular Languages and Finite Automata (AMP)

If $r$ and $s$ are regular expressions, write $r \preceq s$ to mean that the language of strings matching $r$ is contained in the language of strings matching $s$.

(*a*) Show that if $r_1 \preceq s_1$ and $r_2 \preceq s_2$, then $r_1 r_2 \preceq s_1 s_2$. [2 marks]

(*b*) Show that if $r \preceq s$, then $r^* \preceq s^*$. [2 marks]

(*c*) Suppose $s \preceq t$ and $rt \preceq t$. Prove by induction that $r^n s \preceq t$ holds for all $n \geq 0$; deduce that $r^* s \preceq t$. [3 marks]

(*d*) Which of the following instances of the $\preceq$ relation are valid? In each case either give a proof, or specific examples of $r$ and $s$ for which the relation fails to hold. [*Hint:* You may find part (*c*) helpful for some of the proofs.]

(*i*) $r^* \mid s^* \preceq (r \mid s)^*$ [1 mark]

(*ii*) $(r \mid s)^* \preceq r^* \mid s^*$ [1 mark]

(*iii*) $(r^* s^*)^* \preceq (r \mid s)^*$ [2 marks]

(*iv*) $(r \mid s)^* \preceq (r^* s^*)^*$ [2 marks]

(*v*) $(rs \mid r)^* r \preceq r(sr \mid r)^*$ [2 marks]

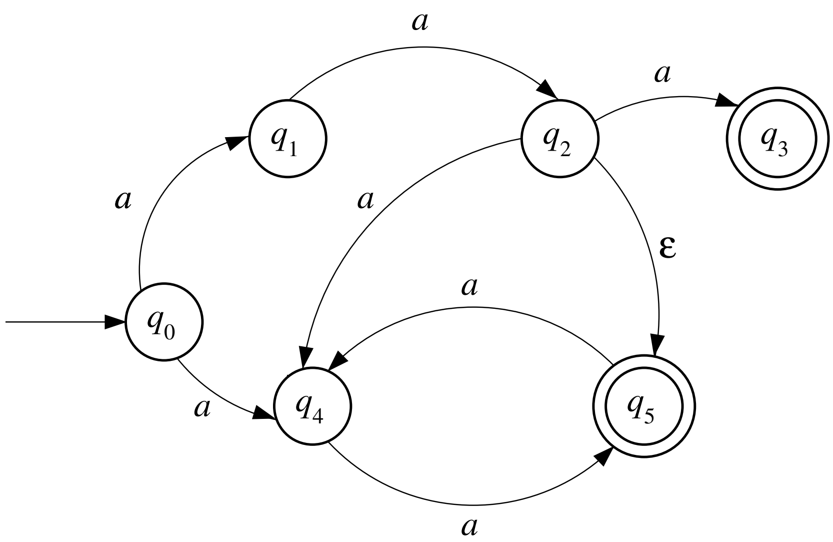
(*e*) Briefly explain why there exists an algorithm for deciding whether or not $r \preceq s$ holds for any given regular expressions $r$ and $s$ (over some fixed alphabet). [5 marks]

# 2011 Paper 2 Question 8

## Regular Languages and Finite Automata

(*a*) Give a regular expression $r$ over the alphabet $\Sigma = \{a, b, c\}$ such that the language determined by $r$ consists of all strings that contain at least one occurrence of each symbol in $\Sigma$. Briefly explain your answer. [5 marks]

(*b*) Let $L$ be the language accepted by the following non-deterministic finite automaton with $\varepsilon$-transitions:




(*i*) Draw a deterministic finite automaton that accepts $L$.

(*ii*) Write down a regular expression that determines $L$.

Briefly explain your answers. [5 marks]

(*c*) Show that if a deterministic finite automaton $M$ accepts any string at all, then it accepts one whose length is less than the number of states in $M$. [5 marks]

(*d*) Is the language

$$\{\, a^n b^\ell a^k \in \{a, b\}^* \mid k \geq n + \ell \,\}$$

regular? Justify your answer. [5 marks]

# 2010 Paper 2 Question 9

## Regular Languages and Finite Automata

($a$) Let $M$ be a finite automaton and let $M'$ be obtained from $M$ by interchanging the collections of accepting and non-accepting states.

($i$) What does it mean for $M$ to be *deterministic*? [2 marks]

($ii$) If $M$ is deterministic, explain why the language accepted by $M'$ is the complement of the language accepted by $M$. [3 marks]

($iii$) Give an example, with justification, to show that the property in part ($ii$) can fail to hold if $M$ is non-deterministic. [2 marks]

($b$) Draw pictures of non-deterministic finite automata with $\varepsilon$-transitions over input alphabet $\{a, b\}$ whose languages of accepted strings are

($i$) $\{a, aa, aaa\}$ [1 mark]

($ii$) all strings not in $\{a, aa, aaa\}$ [3 marks]

($iii$) all strings whose length is divisible by 3 or 5 [3 marks]

($iv$) all strings matching the regular expression $(aa|b)^*(bb|a)^*$ [3 marks]

($v$) all strings not matching the regular expression $(\emptyset^*)^*$ [3 marks]

# 2009 Paper 2 Question 9

## Regular Languages and Finite Automata

Let $L$ be a language over an alphabet $\Sigma$. The equivalence relation $\sim_L$ on the set $\Sigma^*$ of finite strings over $\Sigma$ is defined by $u \sim_L v$ if and only if for all $w \in \Sigma^*$ it is the case that $uw \in L$ if and only if $vw \in L$.

(*a*) Suppose that $L = L(M)$ is the language accepted by a deterministic finite automaton $M$. For each $u \in \Sigma^*$, let $s(u)$ be the unique state of $M$ reached from the initial state after inputting the string $u$. Show that $s(u) = s(v)$ implies $u \sim_L v$. Deduce that for this $L$ the number of $\sim_L$-equivalence classes is finite. [Hint: if $M$ has $n$ states, show that no collection of equivalence classes can contain more than $n$ distinct elements.] [10 marks]

(*b*) Suppose that $\Sigma = \{a, b\}$ and $L$ is the language determined by the regular expression $a^*b(a|b)$. Using part (*a*), or otherwise, give an upper bound for the number of $\sim_L$-equivalence classes for this $L$. [5 marks]

(*c*) Suppose that $\Sigma = \{a, b\}$ and $L = \{a^n b^n \mid n \geq 0\}$. By considering $a^n$ for $n \geq 0$, or otherwise, show that for this $L$ there are infinitely many different $\sim_L$-equivalence classes. [5 marks]

# 2008 Paper 2 Question 8

## Regular Languages and Finite Automata

(*a*) Explain what is a *context-free grammar* and the language it generates.

[4 marks]

(*b*) What does it mean for a context-free grammar to be regular? Given any deterministic finite automaton $M$, describe a regular context-free grammar that generates the language of strings accepted by $M$. [4 marks]

(*c*) Construct a non-deterministic finite automaton with $\varepsilon$-transitions whose language of accepted strings is equal to the language over the alphabet $\{a, b, c\}$ generated by the context-free grammar with non-terminals $q_0$ and $q_1$, whose start symbol is $q_0$ and whose productions are $q_0 \to abq_1$, $q_1 \to \varepsilon$, $q_1 \to q_0$ and $q_1 \to abc$. [4 marks]

(*d*) Is every language generated by a context-free grammar equal to the set of strings accepted by some non-deterministic finite automaton with $\varepsilon$-transitions? Justify your answer. (Any standard results about regular languages you use should be carefully stated, but need not be proved.)

[8 marks]

## Regular Languages and Finite Automata

(*a*) State the *Pumping Lemma* for regular languages. Is every language that satisfies the pumping lemma property a regular language? [5 marks]

(*b*) State, with justification, whether or not each of the following languages is regular. Any standard results you use should be clearly stated, but need not be proved.

(*i*) $L_1 = \{ww \mid w \in \{a\}^*\}$ [3 marks]

(*ii*) $L_2 = \{ww \mid w \in \{a, b\}^*\}$ [3 marks]

(*iii*) $L_3 = \{w_1 w_2 \mid w_1 \in \{a\}^* \text{ and } w_2 \in \{b\}^*\}$ [3 marks]

(*iv*) $L_4 = \{w \mid w \in \{a, b\}^* \text{ and } w \text{ contains the same number of } a\text{s and } b\text{s}\}$ [3 marks]

(*v*) $L_5 = \{w \mid w \in \{a, b\}^*, w \text{ contains the same number of } a\text{s and } b\text{s}, \text{ and that number is no more than } 128\}$ [3 marks]

# 2006 Paper 2 Question 8

## Regular Languages and Finite Automata

(*a*) Suppose that $L_1$ and $L_2$ are regular languages (over the same alphabet $\Sigma$) accepted by deterministic finite automata $M_1$ and $M_2$ respectively. Show that there is a *deterministic* finite automaton $M$ such that for all strings $u$ over $\Sigma$, $M$ accepts $u$ if and only if $u \notin L_1$ or $u \in L_2$. [8 marks]

(*b*) Show that if a deterministic finite automaton $M$ over alphabet $\Sigma$ accepts all strings of length less than the number of states in $M$, then it must accept all strings over $\Sigma$. [4 marks]

(*c*) What does it mean for two regular expressions over an alphabet $\Sigma$ to be *equivalent*? Using parts (*a*) and (*b*), or otherwise, describe an algorithm for deciding equivalence of regular expressions. State carefully any standard results that you rely upon. [8 marks]

# 2005 Paper 2 Question 9

## Regular Languages and Finite Automata

(*a*) Languages $L_1, L_2$ over alphabets $\Sigma_1, \Sigma_2$ are accepted by deterministic finite automata $M_1, M_2$. Show how to construct a deterministic finite automaton $M$ from $M_1$ and $M_2$ that accepts the intersection $L_1 \cap L_2$ of the two languages. What happens if $M_1$ and $M_2$ are non-deterministic? [10 marks]

(*b*) A context-free grammar has a set of terminals $\{0, 1, -\}$, a set of non-terminals $\{N, P\}$, where $N$ is the start symbol, and productions given by the following BNF.

$$\begin{aligned} N &::= 0 \mid P \mid -P \\ P &::= 1 \mid P\,0 \mid P\,1 \end{aligned}$$

(*i*) Give a deterministic finite automaton that accepts the language generated by this context-free grammar. [4 marks]

(*ii*) Give a regular expression that determines the same language. [1 mark]

(*iii*) What is meant by *regular* context-free grammars and what is their connection with regular languages? Is the context-free grammar given above regular? [5 marks]

# 2004 Paper 2 Question 9

## Regular Languages and Finite Automata

(*a*) Prove that if $L$ is a regular language, its complement is also regular. [6 marks]

(*b*) For each of the following languages over the alphabet $\{a, b\}$, state whether or not it is regular and justify your answer.

(*i*) $\{w \mid w \text{ is not a palindrome}\}$

(*ii*) $\{a^k \mid k \text{ is a multiple of } 3\}$

(*iii*) $\{a^k \mid k \text{ is prime}\}$ [14 marks]

# 2003 Paper 2 Question 9

## Regular Languages and Finite Automata

($a$) Let $L$ be the set of all strings over the alphabet $\{a, b\}$ that end in $a$ and do not contain the substring $bb$. Describe a deterministic finite automaton whose language of accepted strings is $L$. Justify your answer. [5 marks]

($b$) Explain what is meant by a *regular expression* $\mathbf{r}$ over an alphabet $\Sigma$ and by the language $L(\mathbf{r})$ determined by $\mathbf{r}$. [6 marks]

If a regular expression $\mathbf{r}$ does not contain any occurrence of the symbol $\emptyset$, is it possible for $L(\mathbf{r})$ to be empty? [2 marks]

Explain why it is always possible, given a regular expression $\mathbf{r}$ over $\Sigma$, to find a regular expression $\sim\mathbf{r}$ with the property that $L(\sim\mathbf{r})$ is the set of all strings over $\Sigma$ that are not in $L(\mathbf{r})$. Any standard results you use should be carefully stated, but need not be proved. [7 marks]

# 2002 Paper 2 Question 9

## Regular Languages and Finite Automata

State, with justification, whether or not each of the following languages over $\Sigma = \{a, b\}$ is regular. Any standard results you use should be clearly stated, but need not be proved:

($a$) $\{a^m b^n \mid m, n \in \mathbb{N}\}$ ; [3 marks]

($b$) $\{a^m b^n \mid m \leqslant n\}$ ; [5 marks]

($c$) $\{a^m b^n \mid m + n \leqslant 4\}$ ; [2 marks]

($d$) $\{w \in \Sigma^* \mid w \notin L\}$, where $L$ is some given language which *is* regular; [4 marks]

($e$) $\{w \in \Sigma^* \mid w \notin L\}$, where $L$ is some given language which *is not* regular; [2 marks]

($f$) some infinite subset of the language given in part ($b$). [4 marks]

# 2001 Paper 2 Question 7

## Regular Languages and Finite Automata

(*a*) Suppose that $L$ is a language over a finite alphabet $\Sigma$ with the property that for each number $\ell \geqslant 1$ there is some string $w$ in $L$ with $length(w) \geqslant \ell$ such that no matter how $w$ is split up into three pieces $w = u_1 v u_2$ with $length(u_1 v) \leqslant \ell$ and $length(v) \geqslant 1$, there is some $n \geqslant 0$ for which $u_1 v^n u_2$ is not in $L$. Prove that $L$ cannot be a regular language. [12 marks]

(*b*) State, with justification, whether each of the following languages over $\Sigma = \{a, b\}$ is regular.

(*i*) $L_1 = \{ww \mid w \in \Sigma^*\}$. [5 marks]

(*ii*) $L_2 = \{wvw \mid v, w \in \Sigma^*\}$. [3 marks]

# 2000 Paper 2 Question 7

## Regular Languages and Finite Automata

For each kind of regular expression over an alphabet $\Sigma$, define the language $L(\mathbf{r})$ of strings matching a regular expression $\mathbf{r}$ of that kind. [4 marks]

Define the language $L(M)$ accepted by a deterministic finite automaton $M$. [2 marks]

Prove that for every deterministic finite automaton $M$ with alphabet of input symbols $\Sigma$ it is possible to construct a regular expression $\mathbf{r}$ over $\Sigma$ satisfying $L(\mathbf{r}) = L(M)$. [10 marks]

Illustrate your proof by constructing such an $\mathbf{r}$ for the deterministic finite automaton with state set $\{0, 1, 2\}$, alphabet of input symbols $\{a, b\}$, initial state 0, accepting states 1 and 2, and next-state function

$$\begin{array}{lll} (0, a) \mapsto 2, & (1, a) \mapsto 1, & (2, a) \mapsto 0, \\ (0, b) \mapsto 1, & (1, b) \mapsto 0, & (2, b) \mapsto 2. \end{array}$$

[4 marks]

# 1999 Paper 2 Question 7

## Regular Languages and Finite Automata

Suppose that $L$ is a language over the alphabet $\{0, 1\}$. Let $L'$ consist of all strings $u'$ over $\{0, 1\}$ with the property that there is some string $u \in L$ with the same length as $u'$ and differing from $u'$ in at most one position in the string. Show that if $L$ is regular, then so is $L'$. [Hint: if $Q$ is the set of states of some finite automaton accepting $L$, construct a non-deterministic automaton accepting $L'$ with states $Q \times \{0, 1\}$, where the second component counts how many differences have been seen so far.] [10 marks]

If a deterministic finite automaton $M$ accepts any string at all, it accepts one whose length is less than the number of states in $M$. Explain why. [5 marks]

State Kleene's theorem about regular expressions and deterministic finite automata. [2 marks]

Describe how to decide for any given regular expression whether or not there is a string that matches it. [3 marks]

# 1998 Paper 2 Question 7

**Regular Languages and Finite Automata**

Explain how the Pumping Lemma is used in proofs that languages are not regular.

[3 marks]

State, with justification, whether each of the following statements is true or false.

(*a*) $\{a^m b^{2n} \mid m \geqslant 0 \text{ and } n \geqslant 0\}$ is regular.

(*b*) $\{a^p b^{2q} \mid p, q \text{ prime}\}$ is regular.

(*c*) No infinite subset of $\{a^n b^n \mid n \geqslant 0\}$ is regular.

(*d*) No infinite subset of $\{ww \mid w \in \{a, b\}^*\}$ is regular.

(*e*) Every finite subset of $\{ww \mid w \in \{a, b\}^*\}$ is regular.

[17 marks]

# 1997 Paper 2 Question 7

## Regular Languages and Finite Automata

Janet and John have been asked to produce a formal design for a piece of sequential hardware. Janet starts her design by setting up a regular grammar that characterises the behaviour needed, while John starts with a regular expression. When they have each separately finished that part of their design they decide that they should check to see whether the languages described by their two formalisms are the same.

Explain how (in a systematic way) they can do this. Standard results that they rely on should be stated explicitly and precisely, and comments about the expected costs of performing the comparison may be useful. [20 marks]

# 1996 Paper 2 Question 8

## Regular Languages and Finite Automata

Show that if $L$ is a regular language then the set of strings in $L$ of odd length is also a regular language. Is the same true of strings of even length? Justify your answer. [8 marks]

If $L$ is regular language let $L'$ be the set of strings in $L$ that are palindromes. Is is possible that $L'$ is regular? Will $L'$ necessarily be regular? Explain your answer with suitable examples and proofs. [6 marks]

It is known that the language *Pal* consisting of all palindromes is not regular. If possible find a regular language $L$ such that $L$ is a subset of *Pal*, or if this is not possible explain why. Similarly either find a regular language $L'$ so that *Pal* is a subset of $L'$, or again explain why this can not be done. [6 marks]

# 1995 Paper 4 Question 3

## Regular Languages and Finite Automata

Define a *regular grammar*. [2 marks]

Define a *regular expression*. [2 marks]

Show that regular grammars and regular expressions characterise the same class of languages. [6 marks]

The syntax of a propositional calculus can be described by the context-free grammar $G = \langle Vn, Vt, P, S \rangle$

$$\begin{aligned}
&\text{where } Vn = \{S\}, Vt = \{not, if, then, and, or, p, q, r\},\\
&\quad P = \{S \to p,\ S \to q,\ S \to r,\\
&\qquad\quad S \to not\ S,\\
&\qquad\quad S \to if\ S\ then\ S,\\
&\qquad\quad S \to S\ or\ S,\\
&\qquad\quad S \to S\ and\ S\}
\end{aligned}$$

Construct a push-down automaton which accepts the set of strings generated by $G$. [10 marks]

# 1995 Paper 3 Question 3

## Regular Languages and Finite Automata

Describe how to derive from any regular expression a deterministic finite automaton describing the same language. [15 marks]

Justify the claim that the resulting automaton does describe the same language. [5 marks]

# 1995 Paper 2 Question 27

## Regular Languages and Finite Automata

Prove or disprove each of the following statements, stating clearly any well known results that you use.

($a$) The set of strings over the alphabet $\{0, 1\}$ that contain exactly twice as many occurrences of 0 as of 1 is a regular language;

($b$) Let $L$ be a regular language over an alphabet $\Sigma$. Then the language consisting of those $u \in \Sigma^*$ such that there is some $v \in \Sigma^*$ with $uv \in L$, is also a regular language;

($b$) Any finite subset of $\{a, b\}^*$ is a regular language;

($d$) For any regular expressions $\mathbf{r}$ and $\mathbf{s}$, the regular expressions $(\mathbf{r}^*\mathbf{s}^*)^*$ and $(\mathbf{r}|\mathbf{s})^*$ always denote the same language.

[20 marks]

# 1992 Paper 4 Question 9

**Formal Languages and Automata**

(a) Let $L$ be the set of all words over the alphabet $\{a, b\}$ that end in $b$ and do not contain the word $aa$. Describe, with justification, a finite deterministic automaton accepting $L$. [6 marks]

Which, if any, of the following regular expressions denotes $L$? Justify your answer in each case.

(i) $(ab|bb|ba)^*b$

(ii) $(b|ab)^*b$

(iii) $(b|ab)^+$

[6 marks]

(b) $L$ and $F$ are languages over an alphabet $\Sigma$, and $F$ is finite. Prove that $L$ is regular if and only if the union $L \cup F$ is regular. You may use any well-known results provided you state them clearly. [8 marks]

# 1991 Paper 4 Question 6

**Formal Languages and Automata**

(a) Prove or disprove each of the following statements. You may use any well-known results provided you state them clearly.

(i) Given regular languages $L_1$ and $L_2$, the set of strings which are in both $L_1$ and $L_2$ forms a regular language.

(ii) The set $\{ww^R | w \in \Sigma^*\}$ is a regular language over the alphabet $\Sigma = \{a, b\}$ (where $w^R$ denotes the reverse of the string $w$).

(iii) The set of strings $w \in \{0, 1\}^*$ for which the number of occurrences of 0 and the number of occurrences of 1 in $w$ are both even forms a regular language.

(b) $M$ is a finite deterministic automaton with $n$ states. Show that the language accepted by $M$ is non-empty if and only if $M$ accepts a string of length $(n - 1)$ or less.

[Hint: consider the shortest word accepted by $M$, if any.]

# 1991 Paper 3 Question 8

**Formal Languages and Automata**

(a) Explain what is meant by a *context-free grammar* and the language generated by it. Write down a context-free grammar over the alphabet with symbols

$$a \ b \ c \ ( \ ) \ \emptyset \ * \ |$$

which generates the set of all regular expressions over the alphabet $\{a, b, c\}$.

(b) What does it mean for a context-free grammar to be *regular*? Given a regular grammar, show how to construct a finite non-deterministic automaton accepting the language generated by it. Illustrate your answer by considering the regular grammar with productions

$$\begin{aligned} I &\to J \\ J &\to abI \\ I &\to bc \end{aligned}$$

where $a, b, c$ are terminals, $I, J$ are non-terminals and $I$ is the start symbol.

## 1990 Paper 6 Question 10

**Formal Languages and Automata**

Explain what is meant by a *regular grammar*. Indicate how it is possible to derive a regular grammar from any regular expression. Illustrate your answer by considering the regular expression $(((aab)^*|aba)a)^*b$

## 1989 Paper 6 Question 12

**Formal Languages and Automata**

Explain what is meant by

(a) a regular expression;

(b) a (deterministic) finite state machine.

Assuming Kleene's theorem (which states that the regular expressions and finite state machines are closely related), describe what is meant by a regular language by relating such languages to both regular expressions and to machines.

State the Pumping Lemma for regular languages.

For each of the following languages either show that the language is regular (for example by showing it would be possible to construct a finite state machine to recognise it) or use the pumping lemma to show that it is not:

(i) the set of all words not in a given regular language $L$;

(ii) all *palindromes* over the alphabet $\{a, b, c\}$ (a palindrome is a word that is unchanged when reversed, for example, *abcbabcba*);

(iii) if $L$ is a regular language, the language which consists of reversals of the words valid in $L$; thus if $L$ contains the word *abcd*, then the reversed language $L^R$ contains *dcba*;

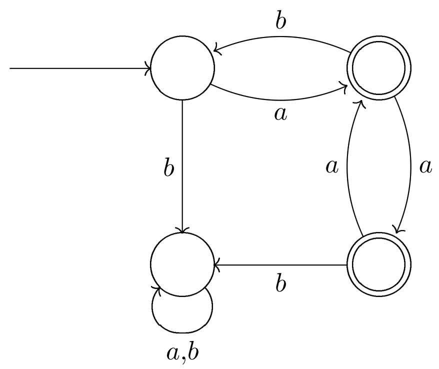
(iv) given regular languages $L$ and $M$, the set of strings that contain within them first a substring that is part of language $L$, then a substring from $M$, arbitrary characters from the alphabet $\{a, b, c\}$ are allowed before, between and after these substrings;

(v) given regular languages $L$ and $M$, the set of strings that contain within them some substring which is part of <u>both</u> $L$ and $M$.

# 1988 Paper 2 Question 3

**Formal Languages and Automata**

(a) Produce a regular expression for the language of this acceptor:




In answering, you may assume Arden's rule which states:
for $R, S, T \subseteq \Sigma^*$, if $\varepsilon \notin S$ and $R = T \cup SR$ then $R = S^*T$.

(b) Prove that if $L$ is a regular language then so is $\{x^R | x \in L\}$, where $x^R$ is the reverse of a string $x$. You may use any well-known results provided you state them clearly.

(Hint: consider the different ways of constructing regular expressions)

# IA Regular Languages and Finite Automata "multi-part" questions

## For 1 mark

1995.2.19 Give a finite deterministic automaton with alphabet of input symbols $\{a, b\}$ that accepts the language denoted by the regular expression $a^*$.

1995.2.20 If $L$ is a regular language over an alphabet $\Sigma$, explain why the complement $\{w \in \Sigma^* | w \notin L\}$ is also a regular language.

1996.2.1(i) Describe in words the strings represented by the regular expression $(aa^*b)^*a^*$.

1996.2.1(j) State the Pumping Lemma for regular languages.

1996.2.1(k) Give a regular grammar that generates the language consisting of even length strings of symbols from the alphabet $\{a, b, c\}$.

1997.2.1(q) Describe the way in which Regular Expressions are constructed.

1998.2.1(s) Give a finite deterministic automaton with alphabet of input symbols $\{a, b\}$ which accepts the language consisting of just the null string $\varepsilon$ and the letter $a$.

1999.2.1(s) What are the differences, if any, between the languages determined by the three regular expressions $\emptyset^*$, $\emptyset(\emptyset^*)$ and $(\emptyset^*)^*$?

## For 4 marks

2000.2.1(b) Draw a picture of a deterministic finite automaton which accepts the language of strings matching $a^*ba$.

2001.2.1(d) Draw a picture of a deterministic finite automaton with set of input symbols $\{a, b\}$ whose language of accepted strings consists of all strings containing an odd number of occurrences of the symbol $a$.

2002.2.1(d) Give a context-free grammar generating the language $\{a^m b^n | m \leq n\}$.

2003.2.1(d) If a deterministic finite automaton accepts any strings at all, why does it accept one whose length is less than the number of states in the automaton?

2004.2.1(d) Draw a state diagram for a deterministic finite automaton that accepts $w \in \{a, b\}^*$ if, and only if, $w$ either begins with $a$ and is of odd length or begins with $b$ and is of even length.