VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

EXAMINATIONS — 2003

TERM TEST

COMP 202

Formal Methods of Computer Science
WITH ANSWERS

Time Allowed: 50 minutes

Instructions: There are **four** (4) questions.
Each question is worth **twenty-five** (25) marks.
Answer **all** the questions.
Show **all** your working.

Question 1.

The following while-program P takes two integers, a and b, and returns integers x and y such that x is the smaller of a and b, and y is the larger of a and b.

begin
  if $a \leq b$ **then**
    $x := a$;
    $y := b$
  else
    $x := b$;
    $y := a$
  fi
end

(a) Write a formal specification (signature, precondition, and postcondition) for this problem. [6 marks]

Input: integers a and b
Output: integers x and y
Precondition: *true*
Postcondition: $\{x, y\} = \{a, b\} \wedge x \leq y$

(b) The following equations may be used to define the semantics of part of the language of while-programs.

$$\mathcal{M}(v := e, \mathcal{S}) = \textbf{update}(\mathcal{S}, v, \mathcal{V}(e, \mathcal{S})) \tag{1}$$

$$\mathcal{M}(T; U, \mathcal{S}) = \mathcal{M}(U, \mathcal{M}(T, \mathcal{S})) \tag{2}$$

$$\mathcal{M}(\textbf{if } c \textbf{ then } T \textbf{ else } U \textbf{ fi}, \mathcal{S}) = \begin{cases} \mathcal{M}(T, \mathcal{S}), & \text{if } \mathcal{V}(c, \mathcal{S}) = \mathit{true} \\ \mathcal{M}(U, \mathcal{S}), & \text{otherwise} \end{cases} \tag{3}$$

(i) *Briefly describe* each of the following:

1. $\mathcal{M}$,
2. $\mathcal{S}$,
3. **update**,
4. $\mathcal{V}$. [8 marks]

$\mathcal{M}$	semantic function for commands: map a store to a new store
$\mathcal{S}$	a store: a mapping from variables to values
update	a function that takes a store $\mathcal{S}$, a variable name v, and a value x, and produces a new store that maps x to v and is otherwise the same as $\mathcal{S}$
$\mathcal{V}$	semantic function for expressions: map a store to a value

(ii) Use the definitions to calculate the final store if the program P above is run with inputs $a = 4$ and $b = 3$. **Show all your working.** [11 marks]

$$
\begin{aligned}
& \mathcal{M}(\textbf{if } a \leq b \textbf{ then } x := a;\ y := b \textbf{ else } x := b;\ y := a \textbf{ fi}, \{(a,4),(b,3)\}) \\
= \;& \mathcal{M}(x := b;\ y := a, \{(a,4),(b,3)\}) \quad \text{since } \mathcal{V}(a \leq b, \{(a,4),(b,3)\}) = \mathit{false} \\
= \;& \mathcal{M}(y := a, \mathcal{M}(x := b, \{(a,4),(b,3)\})) \\
= \;& \mathcal{M}(y := a, \{(a,4),(b,3),(x,3)\}) \\
= \;& \{(a,4),(b,3),(y,4),(x,3)\}
\end{aligned}
$$

Question 2.

(a) *State* Kleene's Theorem. [6 marks]

A language is regular *iff* it is accepted by a FA.
A language is regular *iff* it is accepted by an NFA.
A language is regular *iff* it is described by a regular expression.
A language is regular *iff* it is generated by a regular grammar.

(b) *Outline* the method for constructing an FA from an NFA, such that both machines accept the same language. [9 marks]

See Lecture notes.

(c) Let $M_1 = (Q, \Sigma, \delta, q_0, F)$, where

$$\begin{array}{rcl} Q & = & \{S_0, S_1, S_2, S_3\} \\ \Sigma & = & \{0, 1\} \\ q_0 & = & S_0 \\ F & = & \{S_1\} \end{array}$$

δ	0	1
S_0	$\{S_0, S_1\}$	$\{S_2\}$
S_1	$\{\}$	$\{S_3\}$
S_2	$\{S_1, S_3\}$	$\{\}$
S_3	$\{S_2\}$	$\{S_1\}$

Find an FA which accepts the same language as M_1. [10 marks]

Let $M_1' = (Q', \Sigma', \delta', q_0', F')$, where

$$\begin{aligned}
Q' &= \{\{S_0\}, \{S_0, S_1\}, \{S_2\}, \{S_2, S_3\}, \{\}, \{S_1, S_2, S_3\}, \{S_1\}, \{S_1, S_3\}, \{S_3\}\} \\
\Sigma' &= \Sigma \\
q_0' &= \{S_0\} \\
F' &= \{\{S_0, S_1\}, \{S_1, S_2, S_3\}, \{S_1\}, \{S_1, S_3\}\}
\end{aligned}$$

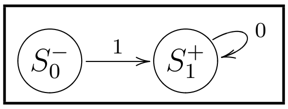
δ'	0	1
$\{S_0\}$	$\{S_0, S_1\}$	$\{S_2\}$
$\{S_0, S_1\}$	$\{S_0, S_1\}$	$\{S_2, S_3\}$
$\{S_2\}$	$\{S_1, S_3\}$	$\{\}$
$\{S_2, S_3\}$	$\{S_1, S_2, S_3\}$	$\{S_1\}$
$\{\}$	$\{\}$	$\{\}$
$\{S_1, S_2, S_3\}$	$\{S_1, S_2, S_3\}$	$\{S_1, S_3\}$
$\{S_1\}$	$\{\}$	$\{S_3\}$
$\{S_1, S_3\}$	$\{S_2\}$	$\{S_1, S_3\}$
$\{S_3\}$	$\{S_2\}$	$\{S_1\}$

Note: $\{\}$ may be omitted.

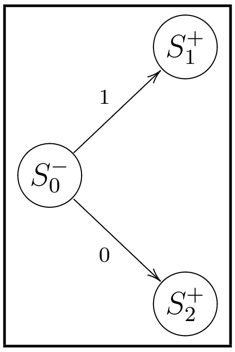
Question 3.

(a) Given an alphabet $\Sigma = \{0, 1\}$, draw NFA with Λ transitions which accept the following languages:

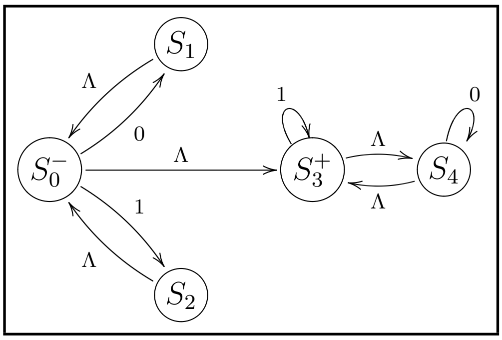
(i) Language($\mathbf{10}^*$) [3 marks]



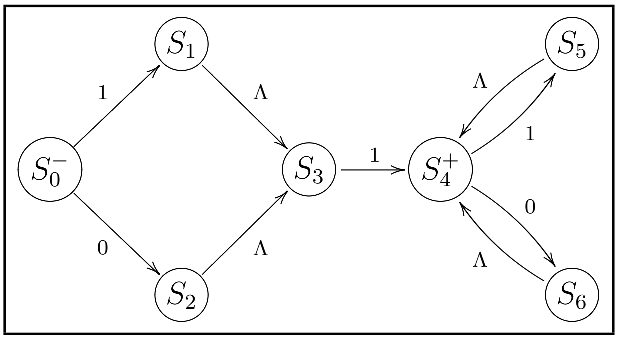
(ii) Language($\mathbf{1} + \mathbf{0}$) [3 marks]



(iii) Language($(\mathbf{0} + \mathbf{1})^*(\mathbf{1}^*\mathbf{0}^*)^*$) [4 marks]



(iv) Language($(\mathbf{1} + \mathbf{0})\mathbf{1}(\mathbf{1} + \mathbf{0})^*$) [4 marks]



(b) Let $M_2 = (Q, \Sigma, \delta, q_0, F)$, where

$$Q = \{S_1, S_2, S_3\}$$

$$\begin{aligned}\Sigma &= \{0,1\}\\ q_0 &= S_1\\ F &= \{S_1, S_3\}\end{aligned}$$

δ	0	1
S_1	S_3	S_2
S_2	S_3	
S_3	S_2	S_1

Find a regular expression which describes the language accepted by M_2. **Show all your working.** [11 marks]

Construct a GNFA $M_2' = (Q', \Sigma', \delta', q_i', q_f')$, where

$$\begin{aligned}Q' &= \{S_i, S_1, S_2, S_3, S_f\}\\ \Sigma' &= \Sigma\\ q_0' &= S_i\\ q_f' &= S_f\end{aligned}$$

δ'	S_1	S_2	S_3	S_f
S_i	Λ	$\emptyset$	$\emptyset$	$\emptyset$
S_1	$\emptyset$	1	0	Λ
S_2	$\emptyset$	$\emptyset$	0	$\emptyset$
S_3	1	0	$\emptyset$	Λ

Now remove states from M_2'.

$M_2'' = (Q'', \Sigma'', \delta'', q_i'', q_f'')$, where

$$\begin{aligned}Q'' &= \{S_i, S_2, S_3, S_f\}\\ \Sigma'' &= \Sigma\\ q_0'' &= S_i\\ q_f'' &= S_f\end{aligned}$$

δ''	S_2	S_3	S_f
S_i	$\emptyset + \Lambda\emptyset^*1$	$\emptyset + \Lambda\emptyset^*0$	$\emptyset + \Lambda\emptyset^*\Lambda$
S_2	$\emptyset + \emptyset\emptyset^*1$	$0 + \emptyset\emptyset^*0$	$\emptyset + \emptyset\emptyset^*\Lambda$
S_3	$0 + 1\emptyset^*1$	$\emptyset + 1\emptyset^*0$	$\Lambda + 1\emptyset^*\Lambda$

Simplifies to:

δ''	S_2	S_3	S_f
S_i	1	0	Λ
S_2	$\emptyset$	0	$\emptyset$
S_3	$0 + 11$	10	$\Lambda + 1$

$M_2''' = (Q''', \Sigma''', \delta''', q_i''', q_f''')$, where

$$\begin{aligned} Q''' &= \{S_i, S_3, S_f\} \\ \Sigma''' &= \Sigma \\ q_0''' &= S_i \\ q_f''' &= S_f \end{aligned}$$

δ'''	S_3	S_f
S_i	$0 + 1\emptyset^*0$	$\Lambda + 1\emptyset^*\emptyset$
S_3	$10 + (0 + 11)\emptyset^*0$	$\Lambda + 1 + (0 + 11)\emptyset^*\emptyset$

Simplifies to:

δ'''	S_3	S_f
S_i	$0 + 10$	Λ
S_3	$10 + (0 + 11)0$	$\Lambda + 1$

$M_2'''' = (Q'''', \Sigma'''', \delta'''', q_i'''', q_f'''')$, where

$$\begin{aligned} Q'''' &= \{S_i, S_f\} \\ \Sigma'''' &= \Sigma \\ q_0'''' &= S_i \\ q_f'''' &= S_f \end{aligned}$$

δ''''	S_f
S_i	$\Lambda + (0 + 10)(10 + (0 + 11)0)^*(\Lambda + 1)$

Question 4.

(a) The pumping lemma tells us that if L is a regular language then there is a number p, such that $(\forall s \in L)(|s| \geq p \Rightarrow s = xyz)$, where:

1. $(\forall i \geq 0)xy^iz \in L$
2. $|y| > 0$
3. $|xy| \leq p$

Outline the proof of the pumping lemma. [8 marks]

The explanation for lectures appears as the next box. The key points that I was looking for were really to say that if the language is regular then there is an FA which accepts it, to mention the pigeon-hole principle, and hence, for sufficiently long strings in the language they can be split and pumped as the PL requires.

We have three things to prove corresponding to conditions 1, 2 and 3 in the pumping lemma. Let:

- $M = (Q, \Sigma, \delta, q_0, F)$ be a FA which accepts L,
- p be the number of states in M (i.e. $p = 2^Q$)
- $s = s_1 s_2 \ldots s_{n-1} s_n$ be a string in L such that $n \geq p$
- $r_1 = q_0$
- $r_{i+1} = \delta(r_i, s_i)$, $1 \leq i \leq n$

Then the sequence $r_1 r_2 \ldots r_n r_{n+1}$ is the sequence of states that the machine goes through to accept s. The last state r_{n+1} is an accepting state.
This sequence has length $n + 1$, which is greater than p. The *pigeonhole principle* tells us that in the first $p + 1$ items in $r_1 r_2 \ldots r_n r_{n+1}$ one state must occur twice.
We suppose it occurs *second* as r_l and *first* as r_j.
Notice: $l \neq j$, and $l \leq p + 1$.
Now let:

- $x = s_1 \ldots s_{j-1}$
- $y = s_j \ldots s_{l-1}$
- $z = s_l \ldots s_n$

So:

- x takes M from r_1 to r_j
- y takes M from r_j to r_j
- z takes M from r_j to r_{n+1}

Hence M accepts $xy^iz, i \geq 0$. Thus we have shown that condition 1 of the pumping lemma holds.
Because $l \neq j$ we know that $|y| \neq 0$. Thus we have shown that condition 2 of the pumping lemma holds.
Because $l \leq p + 1$ we know that $|xy| \leq p$. Thus we have shown that condition 3 of the pumping lemma holds.
Hence the pumping lemma holds.

(b) Give an example of a non-regular language, and explain why it is non-regular.

[8 marks]

Example: $\{1^n0^n|n \geq 0\}$
Proof: We begin the proof by assuming that this is a regular language, so there is some machine N which accepts it.
Hence, by the pumping lemma, there must be some integer k, such that the string 0^k1^k can be pumped to give a string which is also accepted by N.
We let $xyz = 0^k1^k$, and show that $xyyz$ is not in $\{0^n1^n|n \geq 0\}$
There are three cases to consider:

1. y is a sequence of 0s
2. y is a sequence of 0s followed by a sequence of 1s
3. y is a sequence of 1s

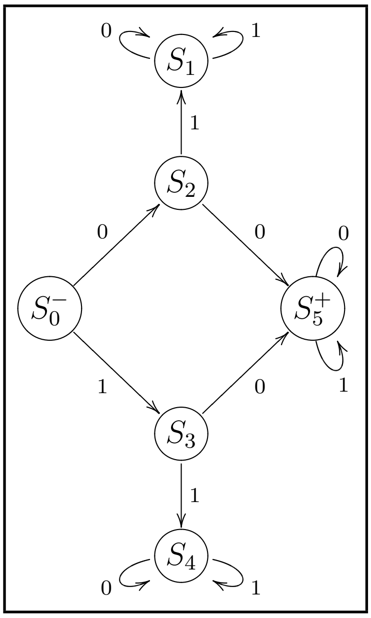
In case 1 $xyyz$ will have more 0s than 1s, and so $xyyz \notin L$.
In case 3 $xyyz$ will have more 1s than 0s, and so $xyyz \notin L$.
In case 2 $xyyz$ will have two occurrences of the substring 01, and so $xyyz \notin L$.
So in each case the assumption that $\{0^n1^n|n \geq 0\}$ is regular leads to a contradiction.
So $\{0^n1^n|n \geq 0\}$ is not regular.

(c) Let $M_3 = (Q, \Sigma, \delta, q_0, F)$, where

$$\begin{aligned} Q &= \{S_0, S_1, S_2, S_3, S_4, S_5\} \\ \Sigma &= \{0, 1\} \\ q_0 &= S_0 \\ F &= \{S_5\} \end{aligned}$$

δ	0	1
S_0	S_2	S_3
S_1	S_1	S_1
S_2	S_5	S_1
S_3	S_5	S_4
S_4	S_4	S_4
S_5	S_5	S_5

(i) Draw M_3. [2 marks]



(ii) Find the smallest FA which accepts the same language as M_3. **Show all your working.** [7 marks]

Split the states of M_3 into equivalence classes, depending on whether we can reach an accepting state using using strings of increasing length:

Λ	$\{S_5\}$	$\{S_0, S_1, S_2, S_3, S_4\}$		
0	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0, S_1, S_4\}$	
1	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0, S_1, S_4\}$	
00	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
10	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
01	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
11	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
000	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
010	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
001	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
011	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
100	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
110	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
101	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$
111	$\{S_5\}$	$\{S_2, S_3\}$	$\{S_0\}$	$\{S_1, S_4\}$

- If you test the strings in different orders (e.g. 11 before 00) there will be minor differences in the table.
- The picture should tell you that this full table need not be written out.
- $\{S_1, S_4\}$ contains no accepting state, and only has transitions into itself, and so can be omitted.

Let $M_3' = (Q', \Sigma', \delta', q_0', F')$, where

$$
\begin{aligned}
Q' &= \{\{S_5\}, \{S_2, S_3\}, \{S_0\}\} \\
\Sigma' &= \{0, 1\} \\
q_0' &= \{S_0\} \\
F' &= \{\{S_5\}\}
\end{aligned}
$$

δ	0	1
$\{S_0\}$	$\{S_2, S_3\}$	$\{S_2, S_3\}$
$\{S_2, S_3\}$	$\{S_5\}$	
$\{S_5\}$	$\{S_5\}$	$\{S_5\}$

For what it is worth, the language accepted is $\mathrm{Language}((\mathbf{1}+\mathbf{0})\mathbf{0}(\mathbf{1}+\mathbf{0})^*)$.

****