Student ID: .....

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS – 2014

END-OF-YEAR

SWEN 224

Formal Foundations of Programming

Time Allowed: THREE HOURS

Instructions: Open Book

You may answer all 5 questions but **only the best 4 answers** will count to your final mark

All questions are of equal value

No Calculators permitted

Question	Topic	Marks	Achieved
1.	Assertions	45	
2.	Verifying Programs	45	
3.	Finite Languages	45	
4.	Regular Languages	45	
5.	Processes	45	
	Total of best 4 questions	180	

Question 1. Assertions [45 marks]

Let A be an array of integers indexed from 0 to $|A| - 1$ and B be an array of integers indexed from 0 to $|B| - 1$. Let x, y and z be integers.

(a) Consider the following assertion:

$$x < y \wedge x = z$$

(i) [1 mark] State in words what the assertion means.

x is smaller than y and equals to z

For each of the following, say whether the above assertion is true or false:

(ii) [1 mark] $x = 2, y = 3, z = 3$

false

(iii) [1 mark] $x = 3, y = 3, z = 3$

false

(iv) [1 mark] $x = 2, y = 3, z = 2$

false

(b) Consider the following assertion:

$$\forall j \bullet 0 \leq j < |A| \rightarrow A[j] > 10$$

(i) [2 marks] State in words what the above assertion means.

All elements in A are larger than 10.

For each of the following arrays, say whether the above assertion is true or false:

(ii) [1 mark] $A = [100, 50, 10]$

false

(iii) [1 mark] $A = []$

true

(c) Consider the following assertion

$$\Big(\exists j \bullet 1 \leq j < |A| \rightarrow A[j-1] = A[j]\Big) \wedge \Big(\forall i \bullet 0 \leq i < |A| \rightarrow A[i] > i\Big)$$

(i) [4 marks] State in words what the above assertion means.

Two consecutive elements in A are equal and all elements in A are greater than their index.

For each of the following arrays, say whether the above assertion is true or false.

(ii) [1 mark] $A = [1,3,3,5]$

true

(iii) [1 mark] $A = [1,2,3,3,7]$ false

false

(iv) [1 mark] $A = []$

false

(v) [1 mark] $A = [10,10,3]$

true

(d) Write a First Order Logic Predicate formalising each of the following statements:

(i) [3 marks] x is less than y and greater than z.

$$y > x > z$$

(ii) [4 marks] A contains at least one element greater than d.

$$\exists i \bullet 1 \leq i \leq |A| \land d < A[i]$$

(iii) [4 marks] If A contains 0 then no element is greater than 10.

$$(\exists i \bullet 0 \leq i < |A| \land 0 = A[i]) \rightarrow (\neg \exists j \bullet 0 \leq j < |A| \land A[i] > 10))$$

(iv) [4 marks] If the two arrays A and B share a common element then these elements are one of the first three elements of each array.

$$\forall i \bullet 0 \leq i < |A| \rightarrow \forall j \bullet 0 \leq j < |B| \rightarrow (A[i] = B[j] \rightarrow (i < 3 \land j < 3))$$

(v) [4 marks] No element in A is in B.

$$\forall z \bullet (\exists : 1 \leq i \leq |A| \bullet z = A[i]) \rightarrow (\neg \exists \bullet 1 \leq j \leq |A| \land z = B[j]))$$

(e) Program contracts.

(i) [5 marks] Write a formal contract, a pre-condition (the requirements the parameters must satisfy) and post-condition (the condition the returned values must satisfy) , for the following program specification.

The program must take three arrays, A, B and C, containing the lengths, widths and heights respectively of a sequence of boxes (cuboids). So $A[i]$, $B[i]$ and $C[i]$ are the length, width and height of box i (the ith box). The program outputs an array, V, containing the volume of the box. Note the volume is defined by $volume = length \times width \times height$.

Pre - $|A| = |B| = |C| \land \forall \bullet i \leq i \leq |A| \rightarrow 0 \leq A[i] \land 0 \leq B[i] \land 0 \leq C[i]$
Post - $|V| = |A| \land \forall \bullet i \leq i \leq |V| \rightarrow V[i] = A[i] \times B[i] \times C[i]$

(ii) [5 marks] Write a formal contract, a pre-condition and post-condition, for the following program specification.

The program inputs, P, an array of pairs. When i is an index of P then the first element of pair $P[i]$ is written as $first(P[i])$ and is a student name. The second element of $P[i]$ is written as $second(P[i])$ and is the integer grade for the student. The program outputs:

1. SB - the set of student names for students that have grade "B", that is their numeric grade is greater than or equal to 70 and less than 75.
2. CB - the number of students that have grade "B".

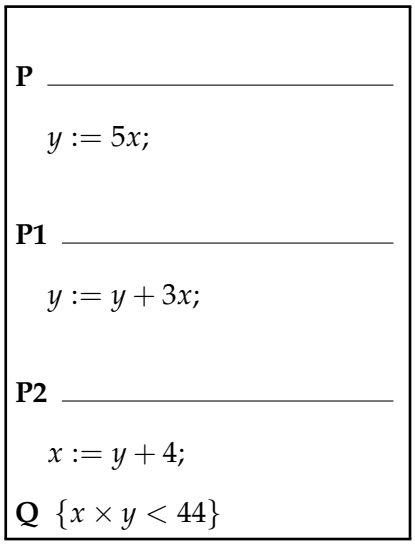
Pre: $\forall \bullet 1 \leq i \leq |A_p| \rightarrow first(P[i]) \in String \wedge second(P[i]) \in Int$
Post: $SB = \{name \bullet \exists \bullet 1 \leq i \leq |P| \wedge P[i] = (name, g) \wedge 70 \leq g < 75\} \wedge CB = |SB|$

Question 2. Verifying Programs [45 marks]

Each question has a program and a post-condition. Use Hoare Logic to compute the missing conditions, simplifying your answers where appropriate.

Variables $x, y, z, w \ldots$ are integers and A is an array of integers.

(a) [5 marks] What are the conditions: **P, P1** and **P2**?



P $\{(8x+4) \times (8x) > 44\} \rightarrow \{16x^2 + 8x) > 11\}$

$y := 5x;$

P1 $\{(y+3x+4) \times (y+3x) > 44\}$

$y := y + 3x;$

P2 $\{(y+4) \times y > 44\}$

$x := y + 4;$

Q $\{x \times y > 44\}$

(b) [5 marks] What are the conditions: **P, P1, P2** and **P3**?

P _____

P1 _____

$x := x - 1;$

Q $\{10 > x + y\}$

P2 _____

$y := -y;$

P3 _____

$x := 5$

Q $\{10 > x + y\}$

If $0 \leq x$ **then** $x := x - 1$**; else** $y := -y; x := 5$**; fi**

Q $\{10 > x + y\}$

P $\{(0 \leq x \wedge 11 > x + y) \vee (\neg(0 \leq x) \wedge y > -5)\}$

P1 $\{10 > x + y - 1\}$

$x := x - 1;$

Q $\{10 > x + y\}$

P2 $\{10 > 5 - y\} \rightarrow \{y > -5\}$

$y := -y;$

P3 $\{10 > 5 + y\}$

$x := 5$

Q $\{10 > x + y\}$

If $0 \leq x$ **then** $x := x - 1$**; else** $y := -y; x := 5$**; fi**

Q $\{10 > x + y\}$

(c) [5 marks] What are the conditions: **P, P1, P2** and **P3**, and the statement **S1**?

P

P1

$y := y + 1;$

P2

$x := 3 - y;$

Q $\{x < 10\}$

P3

S1

Q $\{x < 10\}$

If $(x > 3)$ **then** $y := x + 1; x := 3 - y$**; fi**

Q $\{x < 10\}$

P $=\{(x > 3 \wedge 3 - y < 11) \vee ((\neg x > 3) \wedge x < 10)\} \rightarrow$
$\{(x > 3 \wedge -8 < y) \vee (x \leq 3)\}$

P1 $\{3 - (y + 1) < 10\} \rightarrow$
$\{3 - y < 11\}$

$y := y + 1;$

P2 $\{3 - y < 10\}$

$x := 3 - y;$

Q $\{x < 10\}$

P3 $\{x < 10\}$

S1 skip

Q $\{x < 10\}$

If $(x > 3)$ **then** $y := y + 1; x := 3 - y$**; fi**

Q $\{x < 10\}$

(d) [5 marks] What are the conditions: **P, P1, P2** and **P3**?

P _____

$w := x;$

P1 _____

$x := 8;$

P2 _____

$A[x] := 8;$

P3 _____

$z := A[w];$

Q $\{z > 10\}$

P $\{A\{8 \mapsto 8\}[x] > 10\}$

$w := x;$

P1 $\{A\{8 \mapsto 8\}[w] > 10\}$

$x := 8;$

P2 $\{A\{x \mapsto 8\}[w] > 10\}$

$A[x] := 8;$

P3 $\{A[w] > 10\}$

$z := A[w];$

Q $\{z > 10\}$

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

(e) For a while program with guard **B** and code in the body of the loop **S**:

```
while B do { S }
```

(i) [3 marks] **state the three loop conditions** using the loop invariant **L**, the pre-condition **P** and post-condition **Q**.

1. $P \rightarrow L$
2. $\{L \wedge B\}S\{L\}$
3. $L \wedge \neg B \rightarrow Q$

Consider the following program:

```
int i := 0;
int x := 0;
while (i < z) then {
   x := x + 3;
   i := i + 1;
}
```

(ii) [4 marks] Given the program above, the program pre-condition **Pre** $= \{0 < z \wedge 3z \leq 16\}$ and the program post-condition **Q** $= \{x \leq 16\}$.

Define **L** - the loop invariant and **P** - the loop pre-condition.

$\mathbf{L} = \{i \leq z \wedge x = 3i \wedge 3z \leq 16\}$
$P = \{i = 0 \wedge x = 0 \wedge 0 < y \wedge 0 < z \wedge 3z \leq 16\}$

(iii) [6 marks] Prove your loop invariant, pre-condition and post-condition satisfies the **three loop conditions** (see subquestion (i) above).

$\mathbf{P} \rightarrow \mathbf{L}$
$i = 0 \wedge 0 < z \rightarrow i \leq z$ and
$i = 0 \wedge x = 0 \rightarrow x = 3i$ hence
$\mathbf{P} \rightarrow \{i \leq z \wedge x = 3i \wedge 3z \leq 16\}$
$\{L \wedge B\} S \{L\}$
Pulling **L** back through S we get
$\{i + 1 \leq z \wedge (x + 3) = 3(i + 1) \wedge 3z \leq 16\}$ which implies
$\{i \leq z \wedge x = 3i \wedge 3z \leq 16\}$ which is **L**
$L \wedge \neg B \rightarrow Q$
From $i \leq z \wedge x = 3i \wedge 3z \leq 16 \wedge \neg i < z$
we have $i = z$ and hence $x = 3z$ and finally $x \leq 16$ which is **Q**.

(f) The program below takes two arrays A and B of the same size containing the height and width of rectangles. Rectangle i having height $A[i]$ and width $B[i]$. The program computes a the sum of the areas of the rectangles. The area of a rectangle is $height \times width$ (in the program the area of rectangle i is $A[i] * B[i]$).

```
int i := 0;
int a := 0;
while i < |A| then {
    i := i + 1;
    a := a + (A[i] * B[i]);
}
```

(i) [2 marks] Clearly, the post condition of the program is the post condition of the while loop. From the specification of the program define **Q** - the post-condition of the while loop,

$\mathbf{Q} = \{a = \Sigma_1^{i=|A|}(A[i] \times B[i])\}$

(ii) [4 marks] Define **L** - the loop invariant and **P** - the pre-condition to the loop.

$\mathbf{P} = \{i = 0 \wedge a = 0 \wedge |A| = |B|\}$
$\mathbf{L} = \{i \leq |A| \wedge a = \Sigma_1^{j=i}(A[j] \times B[j])\}$

(iii) [6 marks] Prove your loop invariant, pre-condition and post-condition satisfy the three loop conditions for the given program.

(1) $P \rightarrow L$

Clearly $0 \leq |A|$ and $0 = \Sigma_1^{j=0} A[j] - \Sigma_1^{j=0} B[j]$

With $P \rightarrow i = 0 \wedge s = 0$ we have $P \rightarrow i \leq |A|| \wedge s = (\Sigma_1^{j=i} A[j] - \Sigma_1^{j=i} B[j])$

and hence $P \rightarrow L$

(2) $\{L \wedge B\} S \{L\}$

Pulling L up through S gives

$i + 1 \leq |A| \wedge s + A[i+1] - B[i+1] = \Sigma_1^{j=i+1} A[j] - \Sigma_1^{j=i+1} B[j]$

simplifying $i + 1 \leq |A| \wedge s = \Sigma_1^{j=i} A[j] - \Sigma_1^{j=i} B[j]$

which implies $i \leq |A| \wedge s = \Sigma_1^{j=i} A[j] - \Sigma_1^{j=i} B[j]$

with $|A| = |B|$ we have L and hence

$L \wedge B \rightarrow L$ we have $\{L \wedge B\} S \{L\}$

(3) $L \wedge \neg B \rightarrow Q$

$L \wedge \neg B \rightarrow i \leq |A| \wedge (s = \Sigma_1^{j=i} A[j] - \Sigma_1^{j=i} B[j]) \wedge \neg(i < |A|)$

$\rightarrow i = |A| \wedge (s = \Sigma_1^{j=i} A[j] - \Sigma_1^{j=i} B[j])$

$\rightarrow s = (\Sigma_1^{j=|A|} A[j] - \Sigma_1^{j=|A|} B[j])$

which is **Q**

SPARE PAGE FOR EXTRA ANSWERS

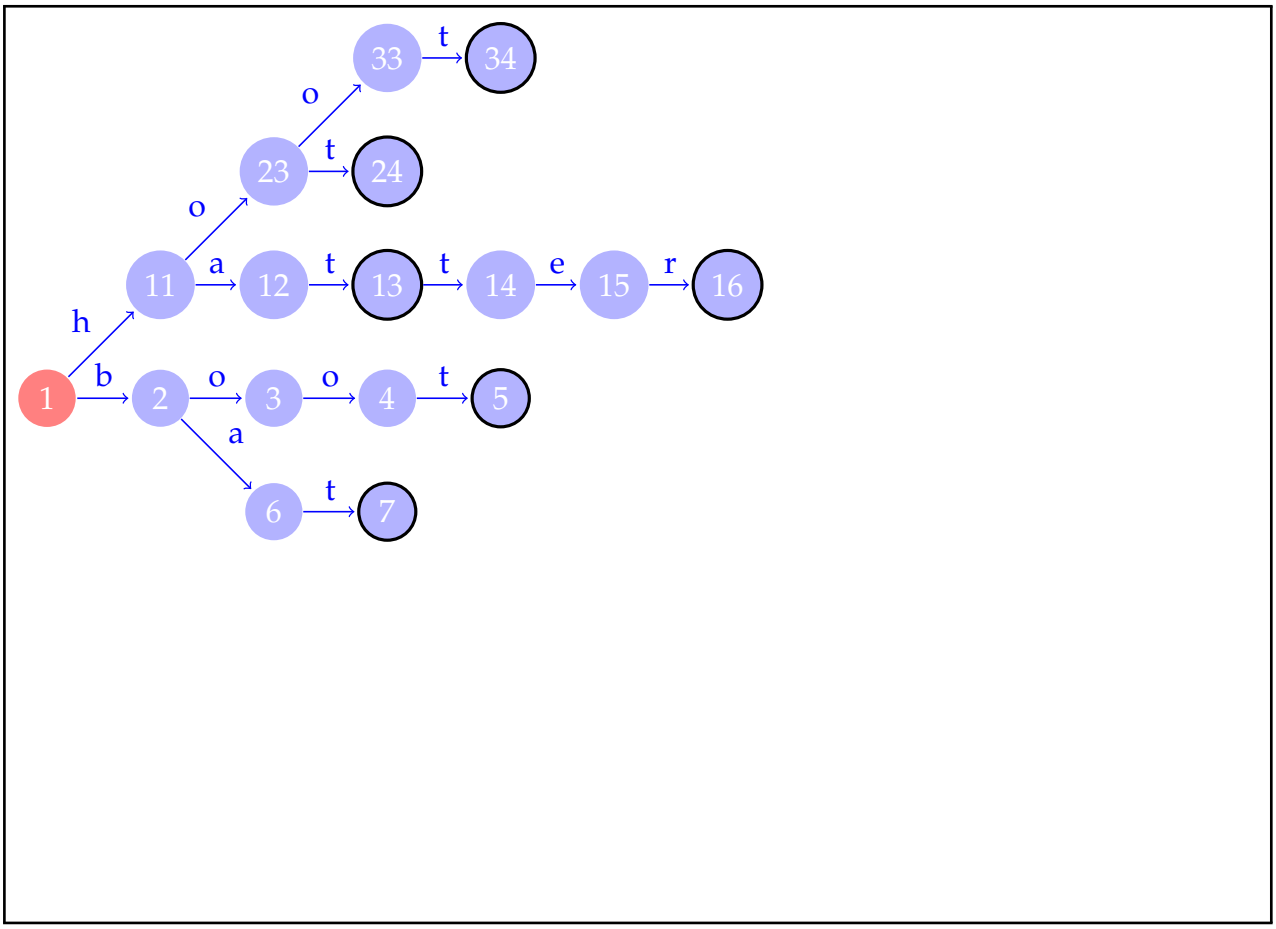
Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 3. Finite Languages [45 marks]

(a) [15 marks] **write down a trie that will recognise all words in the language and only the words in the language** $\{hat, bat, boot, hoot, hot, hatter\}$



(b) [15 marks] From the Trie construct the **language equations**

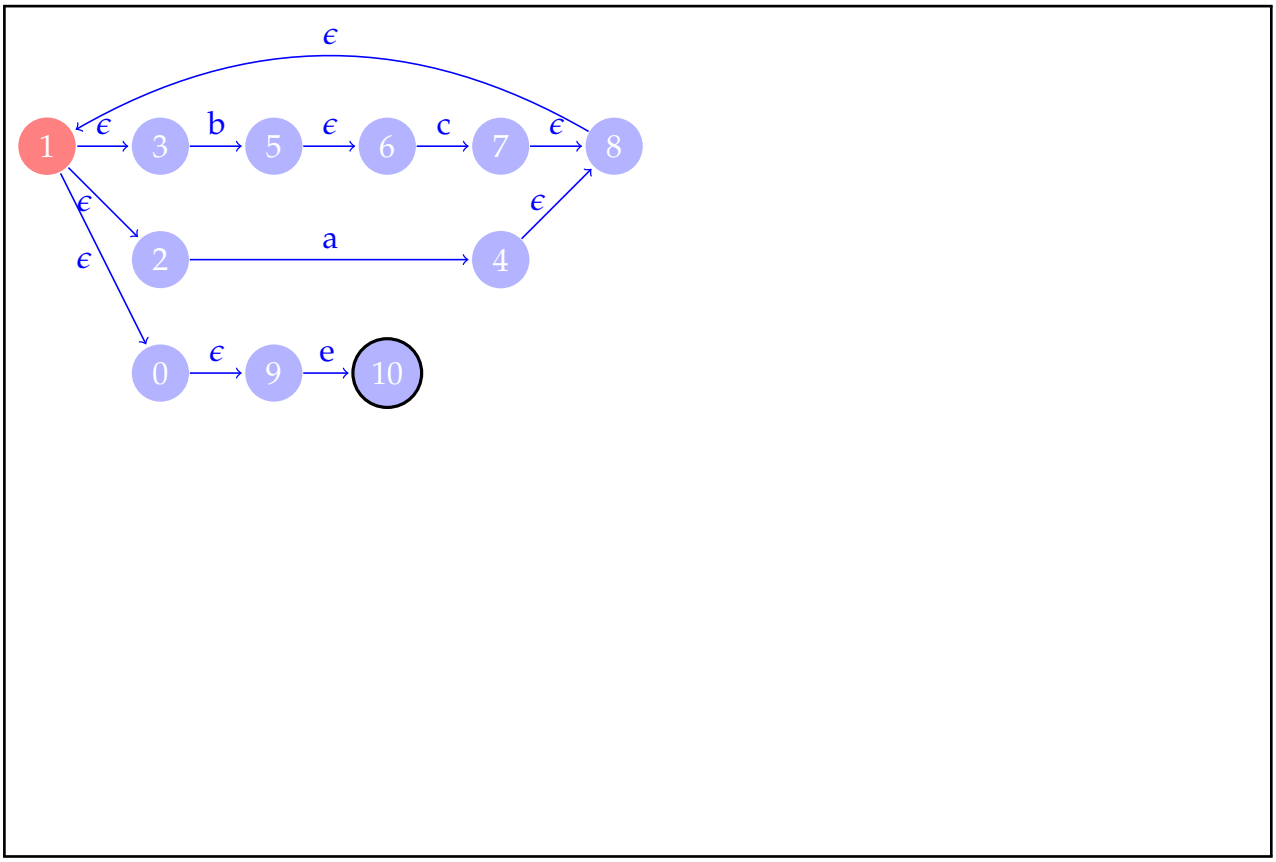
$\mathbb{L}(34) = \mathbb{L}(24) = \mathbb{L}(16) = \mathbb{L}(5) = \mathbb{L}(7) = \{\epsilon\}$ [1]
$\mathbb{L}(15) = r \circ \mathbb{L}(16)$ [2]
$\mathbb{L}(14) = e \circ \mathbb{L}(15)$ [3]
$\mathbb{L}(13) = \epsilon \cup (t \circ \mathbb{L}(14))$ [4]
$\mathbb{L}(12) = t \circ \mathbb{L}(13)$ [5]
$\mathbb{L}(33) = t \circ \mathbb{L}(34)$ [6]
$\mathbb{L}(23) = t \circ \mathbb{L}(24) \cup o \circ \mathbb{L}(33)$ [7]
$\mathbb{L}(11) = a \circ \mathbb{L}(12) \cup o \circ \mathbb{L}(23)$ [8]
$\mathbb{L}(4) = t \circ \mathbb{L}(5)$ [9]
$\mathbb{L}(3) = o\mathbb{L}(4)$ [10]
$\mathbb{L}(6) = t\mathbb{L}(7)$ [11]
$\mathbb{L}(2) = o\mathbb{L}(3) \cup a \circ \mathbb{L}(6)$ [12]
$\mathbb{L}(1) = h \circ \mathbb{L}(11) \cup b \circ \mathbb{L}(2)$ [13]

(c) [15 marks] Solve the equations to reconstruct the language

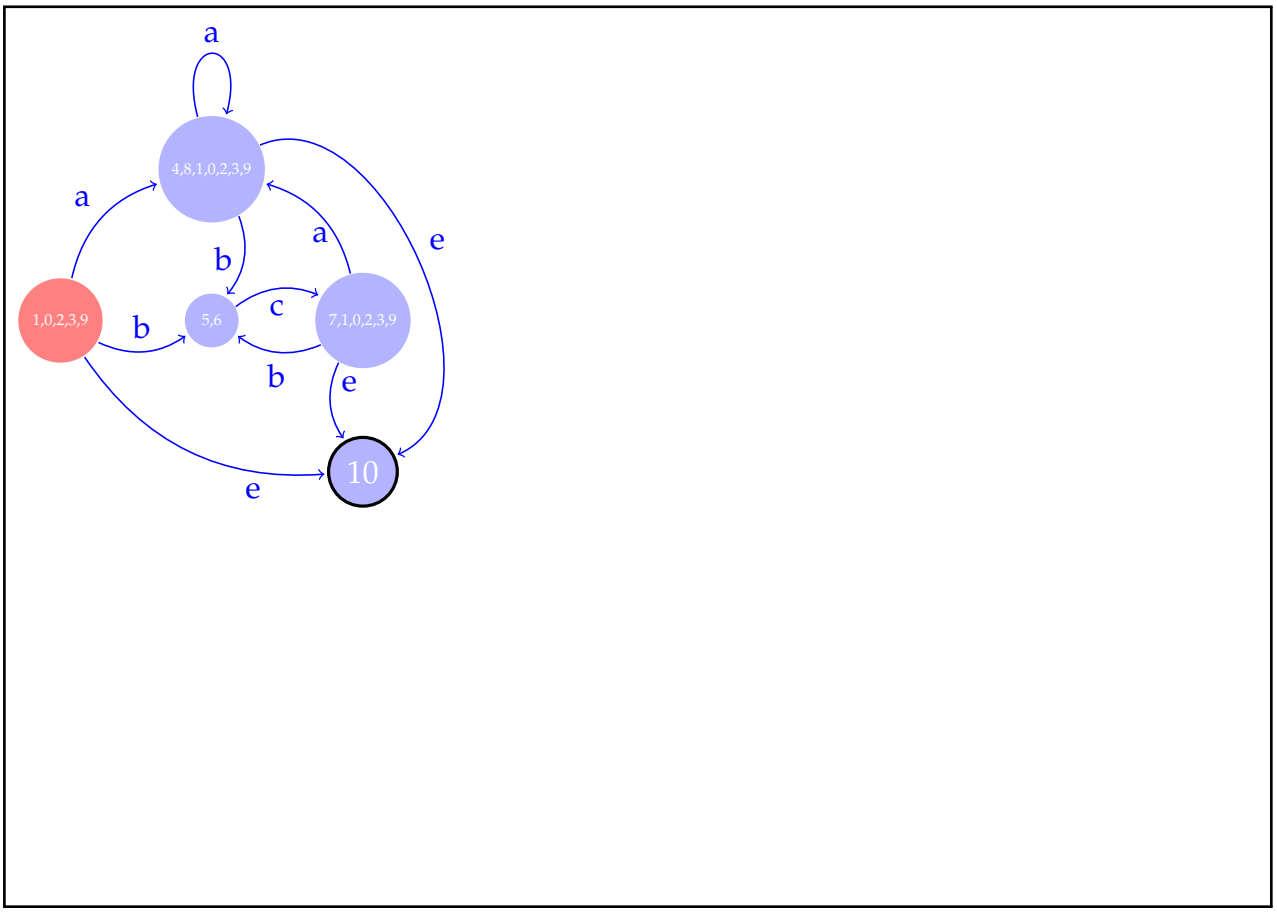
$\mathbb{L}(13) = \epsilon \cup ter$	[4]
Substitution and remove $\circ$	
$\mathbb{L}(23) = t \cup ot$	[7]
$\mathbb{L}(11) = at \circ \mathbb{L}(13) \cup o \circ \mathbb{L}(23)$	[8]
$\mathbb{L}(2) = oot \cup at$	[12]
$\mathbb{L}(1) = h \circ \mathbb{L}(11) \cup b \circ \mathbb{L}(2)$	[13]
Substitution and convert into sets.	
$\mathbb{L}(11) = at \circ \{\epsilon, ter\} \cup o \circ \{t, ot\}$	[8]
$\mathbb{L}(11) = \{at, atter, ot, oot\}$	[8]
$\mathbb{L}(1) = h \circ \{at, atter, ot, oot\} \cup b \circ \{oot, at\}$	[13]
$\mathbb{L}(1) = \{hat, hatter, hot, hoot, boot, bat\}$	[13]

Question 4. Regular Languages [45 marks]

(a) [10 marks] **Construct the Nondeterministic Finite Automata, NFA, from the regular expression** $(a|(b.c))^*.e$



(b) [15 marks] Apply the Subset construction to the NFA in the previous question and display the Deterministic Finite Automata, DFA. **Each node of the DFA must be labeled with a set of NFA nodes.**


a
4,8,1,0,2,3,9
a
e
b
a
1,0,2,3,9
b
5,6
c
7,1,0,2,3,9
b
e
10
e

(c) [10 marks] Write down the equations from the DFA

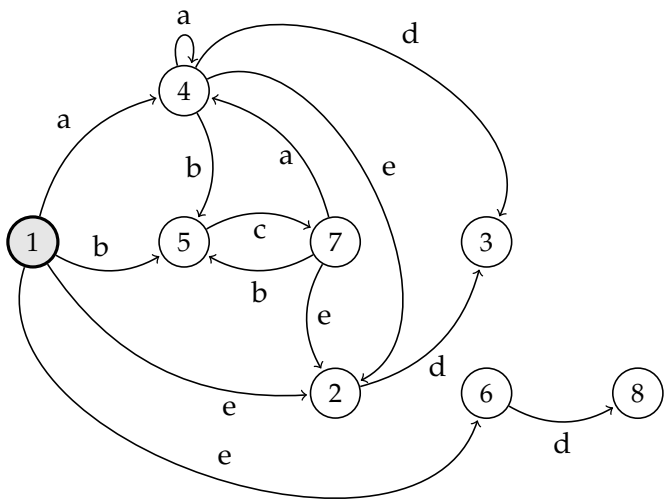
$\mathbb{R}(10) = \epsilon$
$\mathbb{R}(7) = e\mathbb{R}(10)|b\mathbb{R}(5)|a\mathbb{R}(4)$
$\mathbb{R}(5) = c\mathbb{R}(7)$
$\mathbb{R}(4) = e\mathbb{R}(10)|b\mathbb{R}(5)|a\mathbb{R}(4)$
$\mathbb{R}(1) = e\mathbb{R}(10)|b\mathbb{R}(5)|a\mathbb{R}(4)$

(d) [10 marks] Solve these equations to construct a regular expression. Justifying your steps as you go.

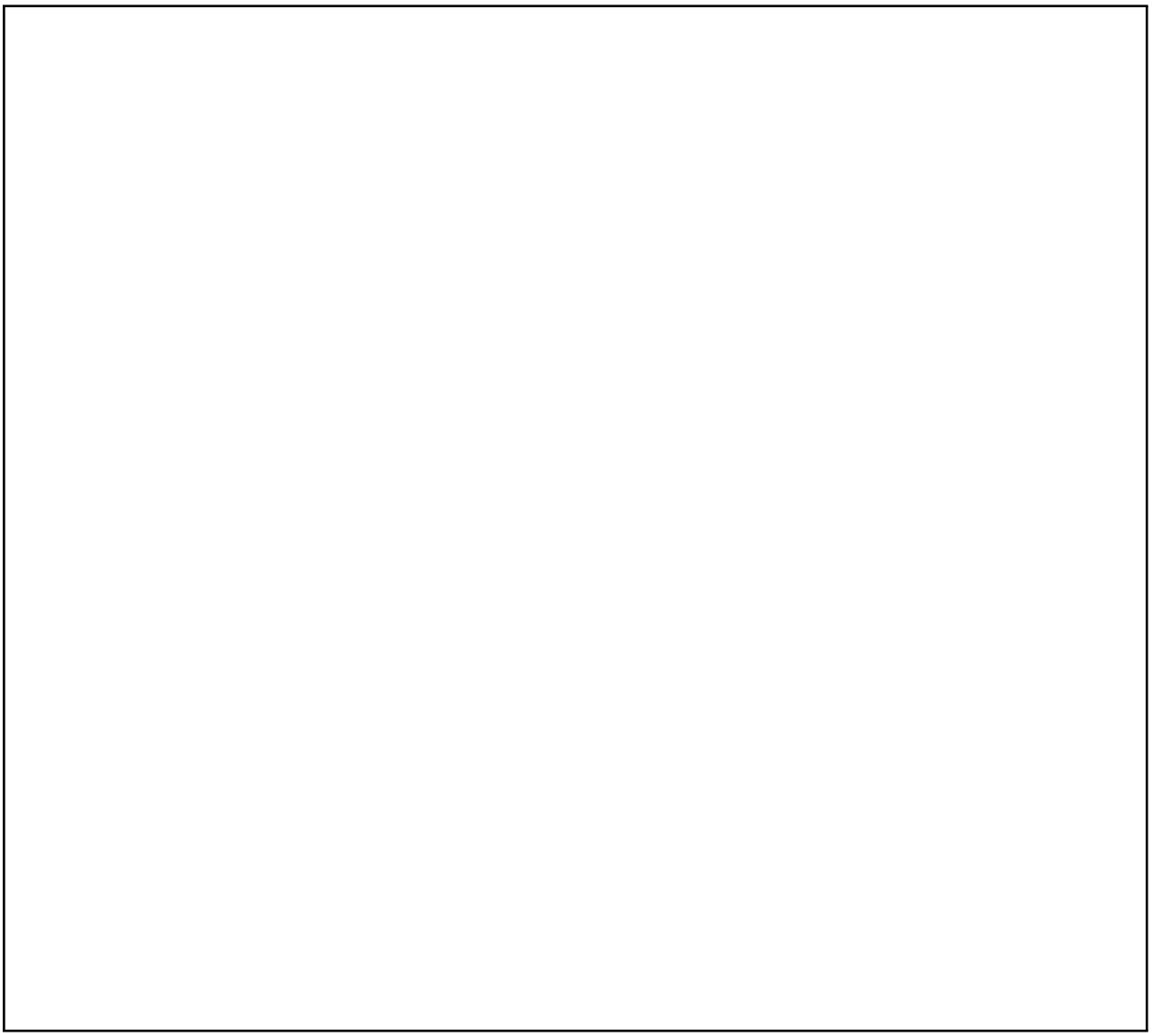
By substitution $\mathbb{R}(1) = e\epsilon|bc\mathbb{R}(7)|a\mathbb{R}(1)$
Transitivity of $=$
$\mathbb{R}(7) = \mathbb{R}(4) = \mathbb{R}(1)$
Hence $\mathbb{R}(1) = e|bc\mathbb{R}(1)|a\mathbb{R}(1)$ by substitution
Hence $\mathbb{R}(1) = e|(bc|a)\mathbb{R}(1)$ from $xz|yz = (x|y)z$
Hence From Ardens $\mathbb{R}(1) = (bc|a)^*e$

Question 5. Processes [45 marks]

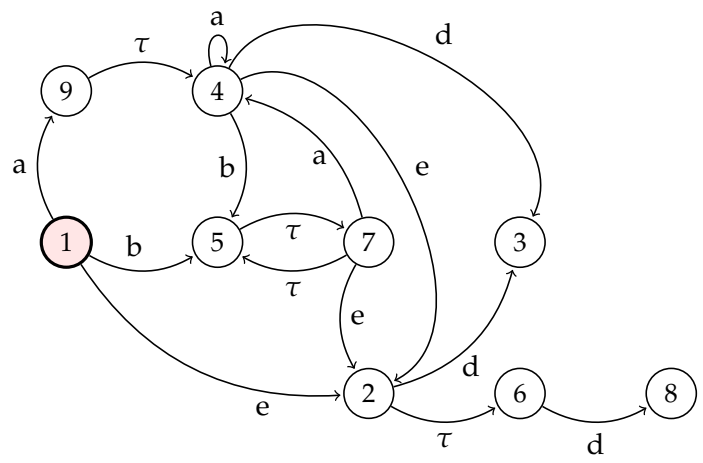
(a) [10 marks] **Build a bisimulation Colouring for the given automata**



(b) [5 marks] **Build an automata from the automata in the previous question by identifying the equivalent nodes**



(c) [15 marks] Add observable events and remove the τ events by applying the abstraction function to the given automata.



SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Build an automata model of a gate controller for a pest proof fence. Name any synchronising events you use SyncA, SyncB, SyncC, ...

(d) [7 marks] The gate has a short corridor with two doors A and B. Build two processes each controlling one of the doors. The controllers can only "talk" to each other, and talk to each other using synchronising events. The controllers must ensure that both doors must not be open at the same time. Use events OpenA, CloseA, OpenB and CloseB.

(e) [8 marks] Refactor your initial model to add a lock down process that includes events Lock and Unlock. When the Lock event is executed the gates are "locked" i.e. can not open. For safety reasons when the Lock event occurs if the last gate open was the B gate then the gates continue to function until the A gate opens once and only once.

* * * * * * * * * * * * * * *