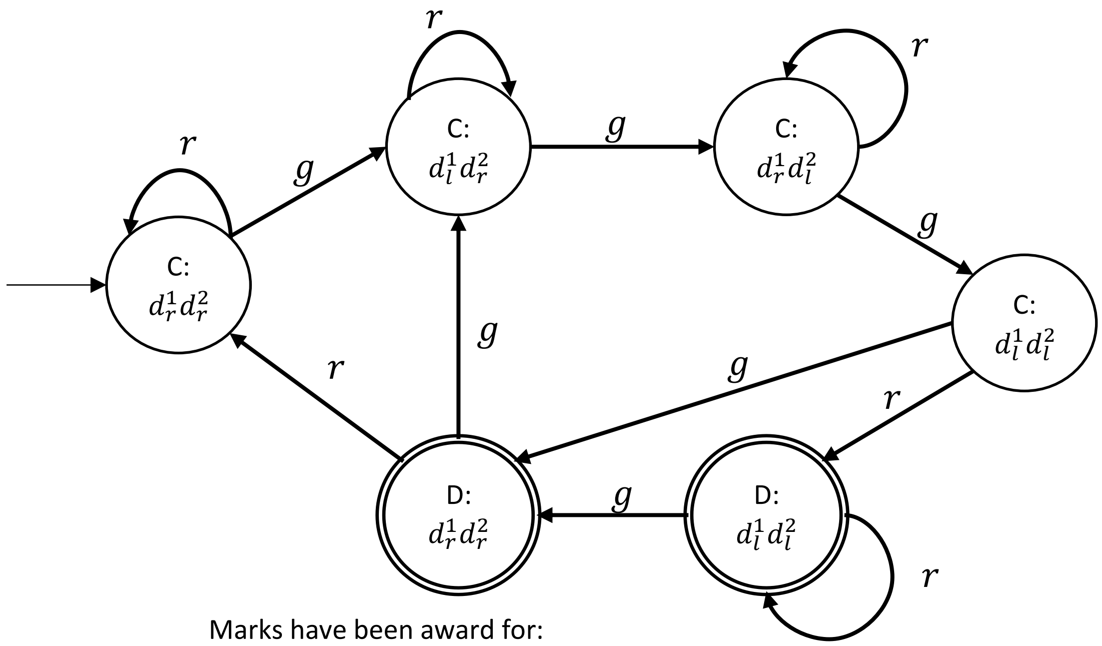
Solutions: CSC2047 Individual Coursework 1

1. a)

   i. The following DFA is an example of a solution that would receive 10 marks



Marks have been award for:

- Showing a valid DFA (3 marks)
  - All states have precisely one transition for each input,
  - The start state is clearly labelled
  - Accept state(s) correctly labelled
- The states represent the state/behaviour of the system (i.e. the exiting at C or D and the current position of the doors) (2 marks)
- The alphabet is $\{g, r\}$. (1 mark)
- The automata models the behaviour of the system correctly. (4 marks)

   ii. The DFA is defined by the 5-tuple

$(Q, \Sigma, \delta, q_0, F)$ (1 mark)

Where $Q = \{C{:}\, d_r^1 d_r^2, C{:}\, d_l^1 d_r^2, C{:}\, d_r^1 d_l^2, C{:}\, d_l^1 d_l^2, D{:}\, d_r^1 d_r^2, D{:}\, d_l^1 d_l^2\}$ (1 mark)

$\Sigma = \{g, r\}$ (1 mark)

$\delta: Q \times \Sigma \to Q$ (3 marks)

Which is defined by the transition diagram

$Q \backslash \Sigma$	g	r
$C{:}\, d_r^1 d_r^2$	$C{:}\, d_l^1 d_r^2$	$C{:}\, d_r^1 d_r^2$
$C{:}\, d_l^1 d_r^2$	$C{:}\, d_r^1 d_l^2$	$C{:}\, d_l^1 d_r^2$
$C{:}\, d_r^1 d_l^2$	$C{:}\, d_l^1 d_l^2$	$C{:}\, d_r^1 d_l^2$
$C{:}\, d_l^1 d_l^2$	$D{:}\, d_r^1 d_r^2$	$D{:}\, d_l^1 d_l^2$
$D{:}\, d_r^1 d_r^2$	$C{:}\, d_l^1 d_r^2$	$C{:}\, d_r^1 d_r^2$
$D{:}\, d_l^1 d_l^2$	$D{:}\, d_r^1 d_r^2$	$D{:}\, d_l^1 d_l^2$

$q_0 = C{:}\, d_r^1 d_r^2$ is the start state (1 mark)

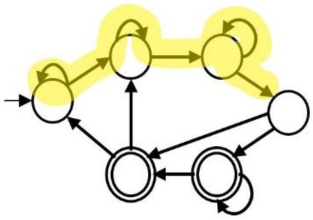
$F = \{D{:}\, d_r^1 d_r^2, D{:}\, d_l^1 d_l^2\}$ (1 mark) b)

i) $gggg$ would be accepted.

ii) $rgrgrgrg$ would be accepted. (1 mark each)

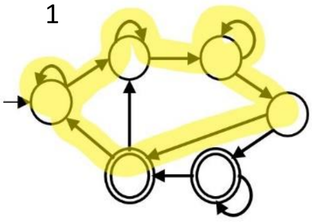
c) There are many different variations of regular expression that can describe the language of the automaton from 1a). Depending on the approach taken to break down the automaton a slightly different regular expression may be generated. Fundamentally, all solutions should cover 5 key components. These components include the initial input sequence required and 4 major "loops" within the automaton which are summarised below:

Initial sequence

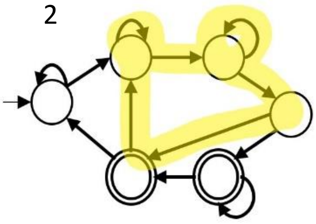


Looping sequences

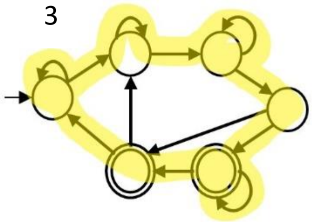
1



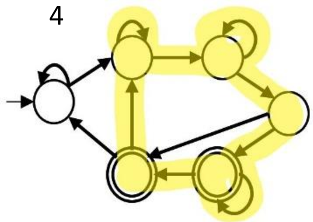
2



3



4



1 mark is awarded for capturing the regular expression for each component. The regular expression below is an example that would have received 5 marks:

$$r^*gr^*gr^*g\big(g \cup \underbrace{gr^+gr^*gr^*g^2}_{\text{Relates to loop 1}} \cup \underbrace{g^2r^*gr^*gr^*g^2}_{\text{Relates to loop 2}} \cup r^+(g \cup \underbrace{gr^+gr^*gr^*grg}_{\text{Relates to loop 3}} \cup \underbrace{g^2r^*gr^*grg}_{\text{Relates to loop 4}})\big)^+$$

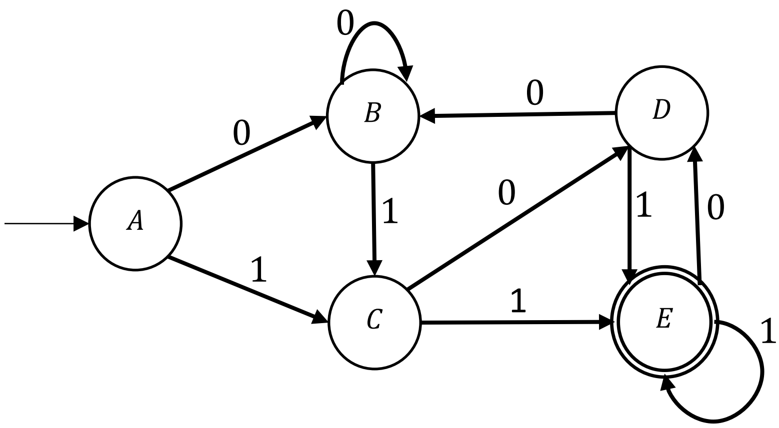
2.

a) The following transition diagram would be generated by following the approach to NFA to DFA conversion outlined in Chapter 7 (Part 2) of the lecture notes.

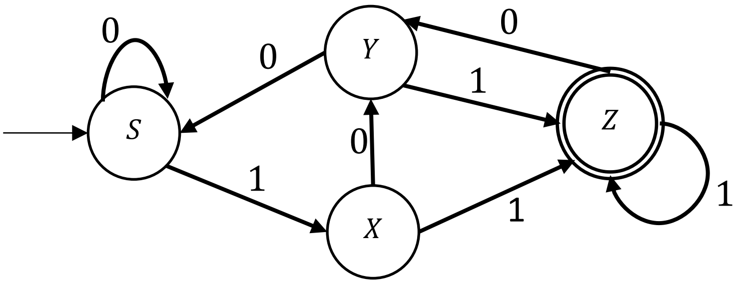
$Q\backslash\Sigma$	0	1
$[q_0]$	$[q_0q_1q_4]$	$[q_0q_1q_2q_4q_5]$
$[q_0q_1q_4]$	$[q_0q_1q_4]$	$[q_0q_1q_2q_4q_5]$
$[q_0q_1q_2q_4q_5]$	$[q_0q_1q_4q_6]$	$[q_0q_1q_2q_3q_4q_5]$
$[q_0q_1q_4q_6]$	$[q_0q_1q_4]$	$[q_0q_1q_2q_3q_4q_5]$
$[q_0q_1q_2q_3q_4q_5]$	$[q_0q_1q_4q_6]$	$[q_0q_1q_2q_3q_4q_5]$

For convenience, the states can be relabelled providing the transition diagram:

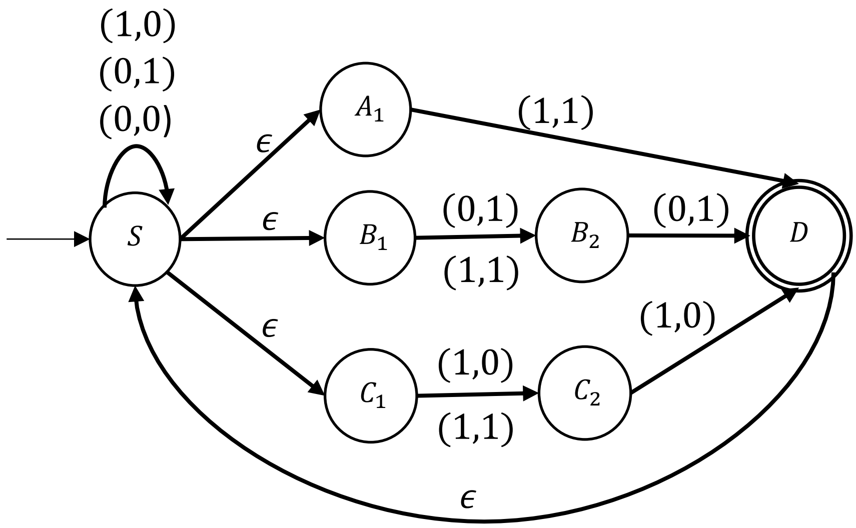
$Q\backslash\Sigma$	0	1
A	B	C
B	B	C
C	D	E
D	B	E
E	D	E



Alternative solutions are also possible. The DFA must recognise the language of all binary strings that end in 11 or 101. Below is an alternative solution that would also receive 10 marks.



b) i)



***The solution must be a valid representation of an NFA (including a singular start state, accept state(s), all states labelled).* *(4 marks)*
***There should be three distinct routes to the accept state(s)* *(2 marks for each = 6)*
***The solution must have an alphabet of tuples (i.e. 2-tuples)* *(2 marks)*
***Note an input alphabet of 00,01,10,11 receives only 1 mark*

ii) The neuron N is modelled by the NFA

$(Q, \Sigma^2, \hat{\delta}, q_0, F)$ such that (1 mark)

$Q = \{q_0, A_1, B_1, B_2, C_1, C_2, D\}$ (1 mark)

$\Sigma = \{0,1\}$ and therefore $\Sigma^2 = \Sigma \times \Sigma = \{(0,0), (0,1), (1,0), (1,1)\}$ (3 marks)

***3 marks also awarded for $\Sigma = \{(0,0), (0,1), (1,0), (1,1)\}$*
***max 1 mark award if alphabet is defined similar to {00, 01, 10, 11}*

q_0 is the start state (1 mark)
$F = \{D\}$ is the accept state (1 mark)
$\hat{\delta}: Q \times \Sigma_\epsilon^2 \to P(Q)$ (3 marks)

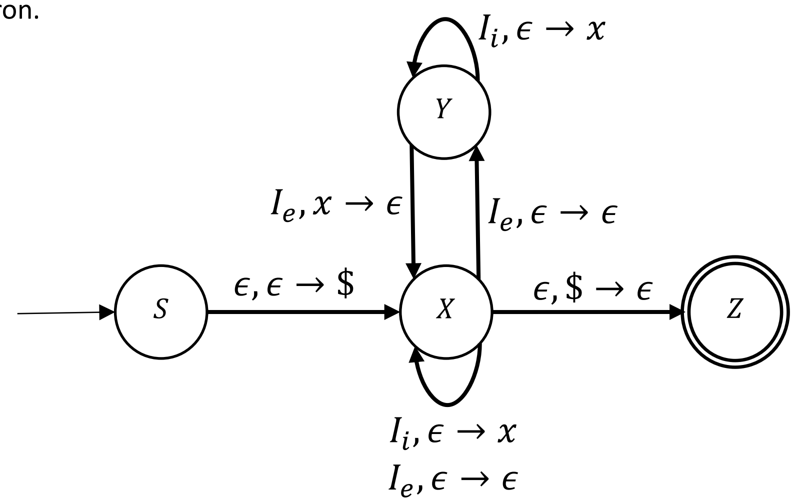
***If ϵ is used in the automaton then the alphabet component must note this inclusion i.e. Σ_ϵ rather than Σ. The alphabet component of the domain must also provide elements that are tuples.*
***The function must be defined correctly i.e. name : domain → codomain*
***The codomain must be the powerset of the states i.e. $P(Q)$ and not Q.*

is defined by the transition diagram (1 mark)

$Q\backslash\Sigma_\epsilon^2$	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$	ϵ
S	$\{S\}$	$\{S\}$	$\{S\}$	$\{D\}$	$\{A_1, B_1, C_1\}$
A_1	$\emptyset$	$\emptyset$	$\emptyset$	$\{D\}$	$\emptyset$
B_1	$\emptyset$	$\{B_2\}$	$\emptyset$	$\emptyset$	$\emptyset$
B_2	$\emptyset$	$\{D\}$	$\emptyset$	$\emptyset$	$\emptyset$
C_1	$\emptyset$	$\emptyset$	$\{C_2\}$	$\emptyset$	$\emptyset$
C_2	$\emptyset$	$\emptyset$	$\{D\}$	$\emptyset$	$\emptyset$
D	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\{S\}$

3. Since the neuron should only reach an accept state when the number of excitatory inputs is high enough **relative to** the number of inhibitory inputs a pushdown automata (PDA) is required. A DFA/NFA cannot keep a record of the number of each input.
(2 marks for identifying that a PDA is required)

The following PDA, represented by the finite state diagram, would model the behaviour of the neuron.



***Must represent a valid PDA i.e. start state, accept state(s), transitions of the form:*

input , pop_character → push_character

States should be labelled appropriately and initial stack character pushed and popped from the stack. *(5 marks)*

***Must model the behaviour of "twice as many" excitatory inputs following inhibitory inputs, i.e. the transitions between X and Y above. Character should be popped from the stack only once when transitioning between X and Y.* *(4 marks)*

***Must model "at least twice as many" excitatory inputs for each inhibitory input already received i.e. the transition for input I_e from state X to state X.* *(2 marks)*

***Must model the behaviour of the automata when an inhibitory input is received. (2 marks)*

4. The language is non-regular. To prove this…

   Assume that the language $L = \{\, a^i b^j \mid i > 2j, \text{for } i, j \in \mathbb{N} \}$ is regular. (1 mark)

   Let k the pump factor (or pumping length) from the pumping lemma. (1 mark)

   Consider the word $a^{2k+1}b^k$ from the language L. Note that $|a^{2k+1}b^k| = 3k + 1$. (3 marks)

   The pumping lemma guarantees that this string can be split in to three pieces xyz such that for any $i \geq 0$ the string xy^iz must also be in L. (1 marks)

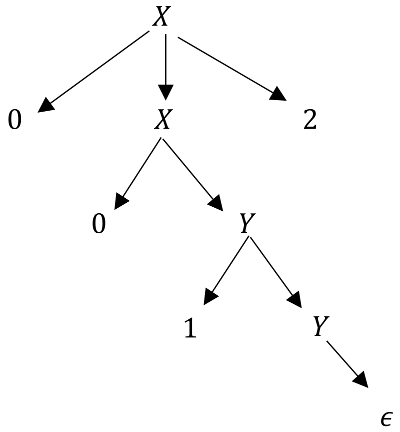
   There are situations to consider when splitting up this word: (3 marks)

   - The string y contains only b's. In this case $|xy| > 2k + 1 > k$ and thus contradicts the condition $|xy| \leq k$ from the pumping lemma.
   - The string y contains a mixture of a's and b's and more precisely, $a^n b^m$ such that $n \leq m$. In this case xy^3z will not be a word from the language L as the number of b's would be $3m + k$ and $2(3m + k) = 6m + 2k \geq 6n + 2k - n$.
   - The string y contains only a's. In this case consider $xy^0z = xz$. In this scenario the number of a's will be at most $2k$ which means the number of a's is not more than twice the number of b's and consequently xy^0z is not in the language L contradicting the pumping lemma.

   Thus a contradiction is unavoidable and therefore our assumption that L is regular must be false. (1 mark)

   ***Note that the argument may be simplified by clearly stating, since $|xy| \leq k$ according to the pumping lemma, the first two cases can be quickly eliminated. This must be made clear for a shorter argument to be accepted.*

5. a) ***A valid parse tree must be shown. 1 mark awarded for each correct layer of the parse tree (ignoring the root).*



b) $\{0^a 1^b 2^c \mid a, b, c \in \mathbb{N}, a = c + 1\}$

*** 1 mark for correct structure of set i.e.* { x | condition }

*** 1 mark for noting that words from this language are in the form 0..1…2*

*** 1 mark for noting that the indices are* $\mathbb{N}$ *(or* ≥ 0*)*

***2 marks for noting that the number of 0's must be one more than the number of 2's and that there is no restriction on the number of 1's.*

c) Create a new start state: (1 mark)

$$S \to X$$
$$X \to 0X2|0Y$$
$$Y \to 1Y|\epsilon$$

Remove ϵ transitions: (2 marks)

$$S \to X$$
$$X \to 0X2|0Y|0$$
$$Y \to 1Y|1$$

"Break the chains": (2 marks)

$$S \to X$$
$$X \to 0X_1|0Y|0$$
$$X_1 \to X2$$
$$Y \to 1Y|1$$

Remove unit rules: (1 mark)

$$S \to 0X_1|0Y|0$$
$$X \to 0X_1|0Y|0$$
$$X_1 \to X2$$
$$Y \to 1Y|1$$

Convert the remaining rules into CNF format and add additional variables for terminals: (2 marks)

$$S \to V_0X_1|V_0Y|V_0$$
$$X \to V_0X_1|V_0Y|V_0$$
$$X_1 \to XV_2$$
$$Y \to V_1Y|V_1$$
$$V_0 \to 0$$
$$V_1 \to 1$$
$$V_2 \to 2$$