ASSIGNMENT/ASSESSMENT ITEM COVER SHEET

Student Name: Nabeel | **Majid**

FIRST NAME | **FAMILY / LAST NAME**

Student Number: 3 2 8 7 0 6 0 Email: c3287060@uon.edu.au

Course Code	**Course Title**
c o m p 2 2 7 0	**Theory of Computation**
(Example) A B C D 1 2 3 4	*(Example)* Intro to University

Campus of Study: Callaghan (eg Callaghan, Ourimbah, Port Macquarie)

Assessment Item Title: assignment2 Due Date/Time: 14/05/23

Tutorial Group (If applicable): Word Count (If applicable):

Lecturer/Tutor Name: Nasim

Extension Granted: (•) Yes ( ) No Granted Until: 17/05/23

Please attach a copy of your extension approval

NB: STUDENTS MAY EXPECT THAT THIS ASSIGNMENT WILL BE RETURNED WITHIN 3 WEEKS OF THE DUE DATE OF SUBMISSION

Please tick box if applicable

☐ *Students within the Faculty of Business and Law, Faculty of Science and Information Technology, Faculty of Engineering and Built Environment and the School of Nursing and Midwifery:*
I verify that I have completed the online Academic Integrity Module and adhered to its principles

☐ *Students within the School of Education:*
"I understand that a minimum standard of correct referencing and academic literacy is required to pass all written assignments in the School of Education; and I have read and understood the School of Education *Course Outline Policy Supplement,* which includes important information related to assessment policies and procedures.

I declare that this assessment item is my own work unless otherwise acknowledged and is in accordance with the University's academic integrity policy available from the Policy Library on the web at http://www.newcastle.edu.au/policylibrary/000608.html
I certify that this assessment item has not been submitted previously for academic credit in this or any other course. I certify that I have not given a copy or have shown a copy of this assessment item to another student enrolled in the course.

I acknowledge that the assessor of this assignment may, for the purpose of assessing this assignment:
- Reproduce this assessment item and provide a copy to another member of the Faculty; and/or
- Communicate a copy of this assessment item to a plagiarism checking service (which may then retain a copy of the item on its database for the purpose of future plagiarism checking).
- Submit the assessment item to other forms of plagiarism checking.

DATE STAMP HERE

I certify that any electronic version of this assessment item that I have submitted or will submit is identical to this paper version.

Turnitin ID:
(if applicable)

Insert this way

Signature: ______________________ Date: 17/05/23

To copy and paste the completed form into another document use the `snapshot' tool.

Print Form

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Problem 1

(a) double($a^* b^*$) is an empty ~~str~~ string because in $s = xwwy$, w can either be a, b or ε hence 'a' and 'b' cannot be adjacent so it has to be empty, double $(a^* b^*) = \{\varepsilon\}$

(b) doubles$(ab)^*$

The Kleene star of this language means zero or more (ab) so its double is (ab).

(C) Yes INF is closed under doubles. We Know if a language is closed under that double if we apply that operation and it we will give answer already in the string. There are two consective value of w and L is infinite so double(L) will also be infinite

(d) No, Regular languages are not closed under doubles. Machine might have to immediatly go through the part of language already been read. Therefore It can be infinite number and Regular language has to be finite.

Problem 2

(a) $L = \{0^m 1^n : m = 3n+2, n \geq 0\}$

In this we have to maintain equation of $3n+2$ and $0^m 1^n$ mean 0 are fallowed by 1.
We have to generate Multiple of 3 of '0' compare to '1' and 2 more '0' before 1.
So,
if $n = 0$ we generate 2 zero's and rest we generate 3 zero's fallowed

$$\boxed{\begin{array}{l} S \rightarrow 00T \\ T \rightarrow 000T \mid 1T \mid \varepsilon \end{array}}$$

Problem#2

(b) $L = \{w \in \{0,1\}$

Ans: In this Language, '0' always ~~or~~ comes in Pair with '1', but ~~one~~ '1' can come by itself e.g 1,

$S \rightarrow TS \mid \varepsilon$
$T \rightarrow 0U \mid 1$
$U \rightarrow 1T \mid 1$.

(c) $L = \{w \in (L_1 L_2)^*$

We need rules for L_1 and L_2 and a S.

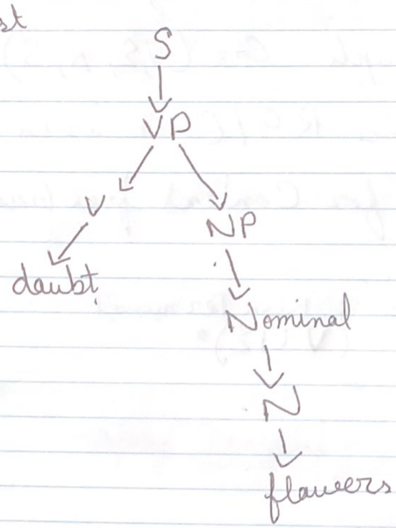
$S \rightarrow L_1 L_2 S \mid \varepsilon$
$L_1 \rightarrow 0 L_1 0 \mid 1 L_1 1 \mid \varepsilon$
$L_2 \rightarrow a L_2 a \mid b L_2 b \mid \varepsilon$

Parse Trees

Problem 3

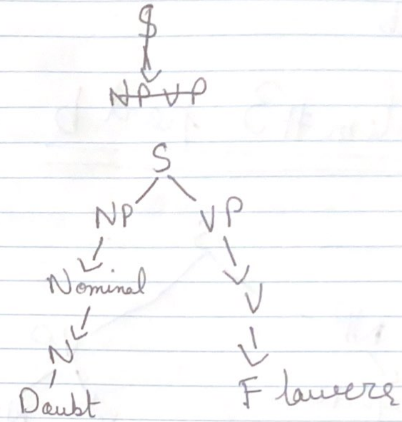
(a) Doubt Flower

First



(a)

2nd Parse Tree

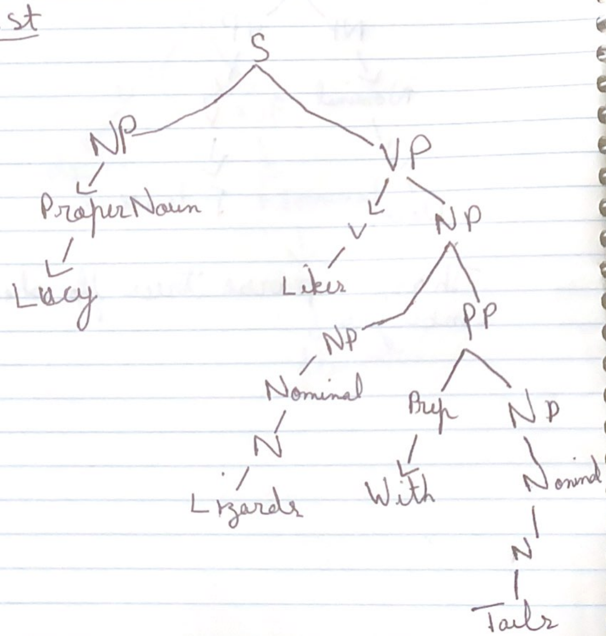


These 2 parse trees ~~flo~~ show ambiguity

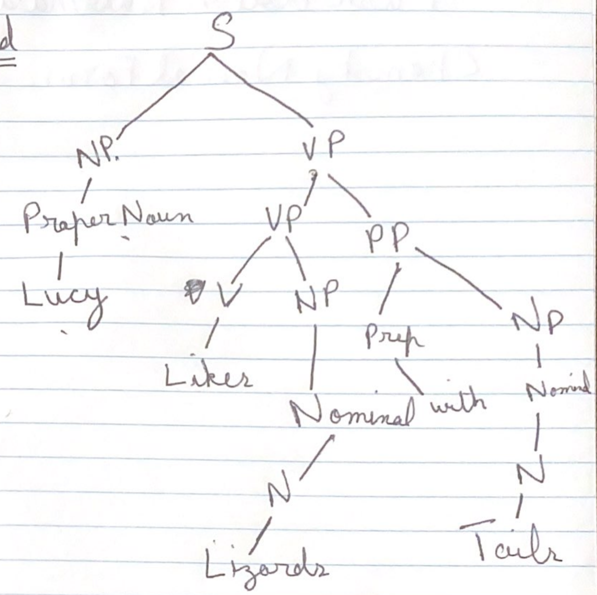
~~Problem#3~~
~~Parb~~

Problem #3 part b

1st



2nd



Above two graphs show ambiguity

Problem # 4

Chomsky normal form.

$S \to XyX$

$X \to YYYY$

$X \to Z$

$Y \to xY$

$Y \to \varepsilon$

$Z \to zZ$

$Z \to z.$

To remove episilon production or Null production

1st. Nullable: X, Y.

$S \to XyX \mid y \mid Xy \mid yX$

$X \to YYYY \mid YYY \mid YY \mid Z$

$Y \to xY \mid x$

$Z \to zZ$

$Z \to z.$

Q#4

2. Removing unit production

$S \to X_y X \mid y X \mid X y \mid y$

$X \to YYY \mid YY \mid x \mid z$

$Y \to xY \mid x$

$Z \to zZ \mid z.$

3. Removing Mixed production

Adding rules

$T_y \to y$, $T_x \to x$, $T_z \to z$.

$S \to \cancel{X} T_y X \mid T_y X \mid X T_y \mid y.$

$X \to YYY \mid YY \mid x \mid z.$

$\cancel{Y \to T_x Y}$ $Y \to T_x Y \mid x$

$Z \to T_z Z \mid z.$

4. Removing long production

$S \to M_1 X \mid T_y X \mid X T_y \mid y$

$X \to M_2 Y \mid YY \mid x \mid z$

$Y \to T_x Y \mid x$

$Z \to T_z Z \mid z$

$Ty \longrightarrow y$

$Tz \longrightarrow z$

$Tx \longrightarrow x$

$M_1 \longrightarrow X \; Ty$

$M_2 \longrightarrow Y Y$

Problem #5

$K = q_0, q_1, q_2,$

$\Sigma = a, b$

$\Gamma = A.$

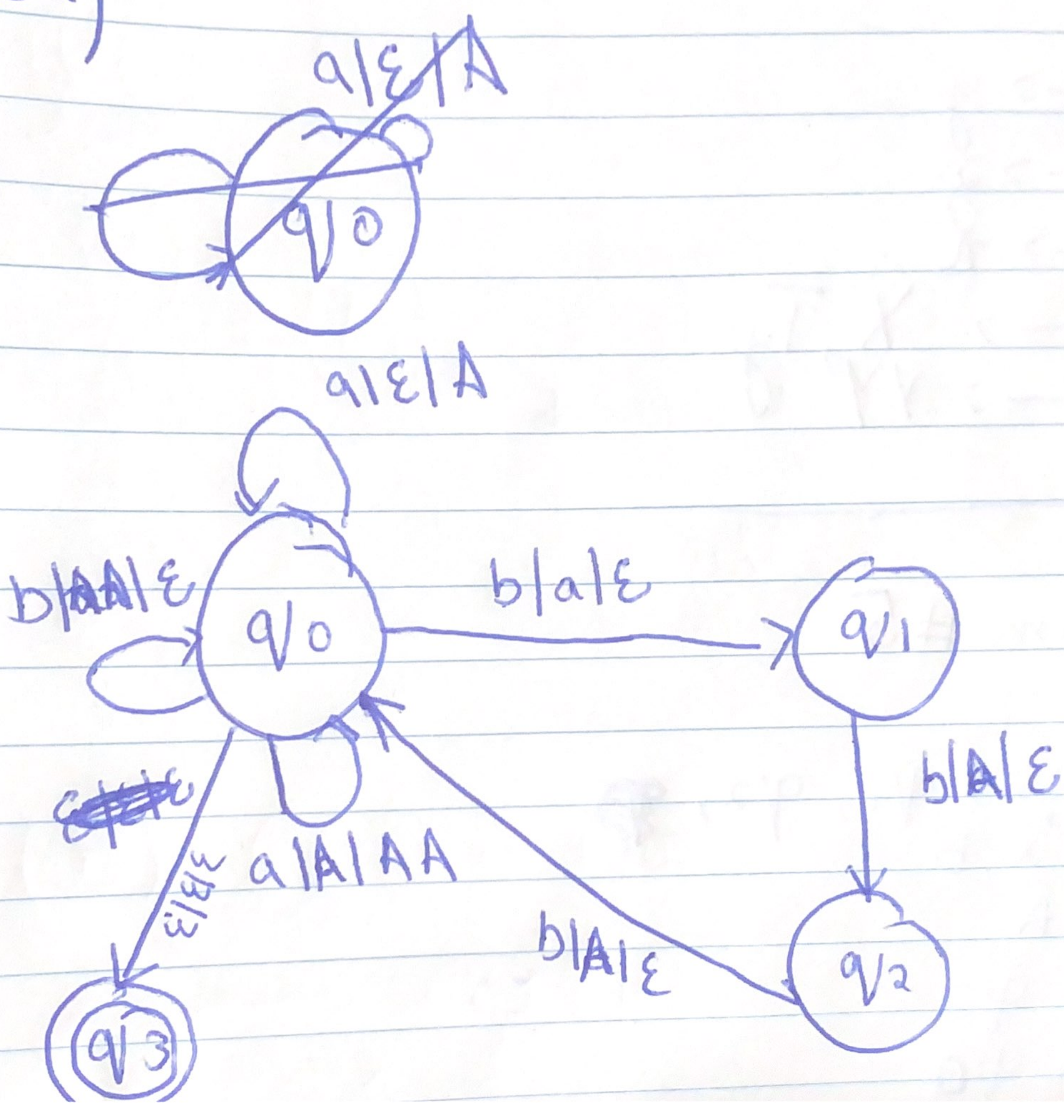
$S = q_0$

$A = q_0$

$\Delta =$

1. $(q_0, a, \varepsilon) = \{(q_0, A)\}$
2. $(q_0, a, A) = \{(q_0, AA)\}$
3. $(q_0, b, AA) = \{(q_0, \varepsilon)\}$
4. $(q_0, b, A) = \{(q_1, \varepsilon)\}$
5. $(q_1, b, A) = \{(q_2, \varepsilon)\}$
6. $(q_2, b, A) = \{(q_0, \varepsilon)\}$
7. ~~$(q_0, \varepsilon, \varepsilon) = \{(q_3, \varepsilon)\}$~~

a)



5(b)

$(q_0, a, \varepsilon) = \{q_0, A\}$

$(q_0, a, A) = \{q_0, AA\}$

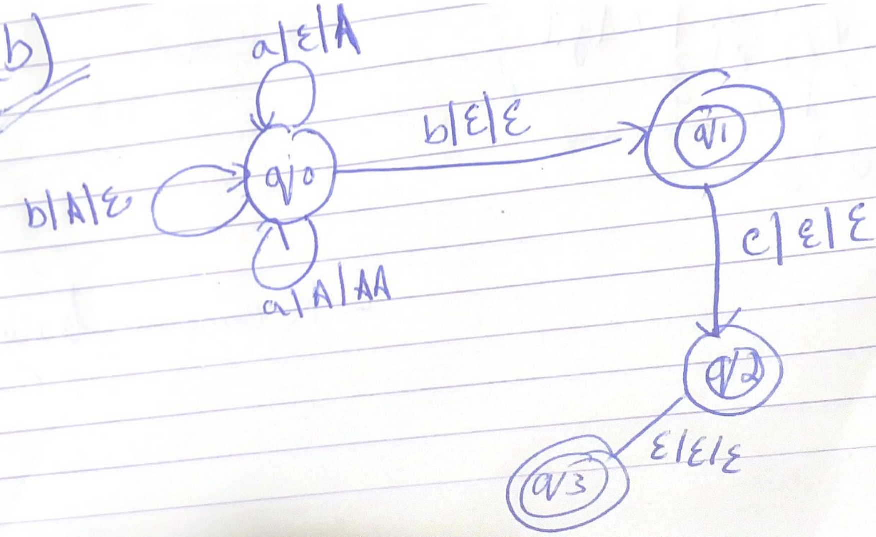
$(q_0, b, A) = \{q_0, \varepsilon\}$

$(q_0, b, \varepsilon) = \{q_1, \varepsilon\}$

$(q_1, c, \varepsilon) = \{q_2, \varepsilon\}$

$(q_2, \varepsilon, \varepsilon) = \{q_3, \varepsilon\}$

5 (b)



Q6

(a) L(M) —> This PDA accept this language if it has same or more number of 'b' as there are 'a'. It start with 'a' and fallowed by same or more 'b'.

6(b)

Content-free-Grammer

$S \longrightarrow aSb \mid T$

$T \longrightarrow bT \mid b$

(c) No, M is not deterministic, because for deterministic PDA There is atmost one clear path. here M has Multiple path.

(d) To Prove M is determinitis

$K = q_0, q_1$
$\Sigma = a, b$
$\Gamma = A$
$S = q_0$
$A = q_0, q_1$
$\Delta = (q_0, a, \varepsilon) = \{q_0, A\}$
$= (q_0, a, A) = \{q_0, AA\}$
$(q_0, b, A) = (q_1, \varepsilon\}$
$(q_1, b, A) = (q_1, \varepsilon\}$
$(q_1, b, \varepsilon) = \{q_1, \varepsilon\}$

Q7 (a)

L_2 is a superset of L_1 and L_2 is content free and not regular

We Know $a^n b^n$ is not regular

$L_2 = \{a^n b^n : n \geq 0\}$

$L_1 = \{a^n\}$

Here, L_1 is the set of all possible strings ~~of 'a'~~ consisting of 'a'. We can say from $L_1 = \{a^n\}$ that it is a regular language because we can design FSM or for it and its reguler expression would be a^*.

L_2 is the language of equal number of 'a' followed by equal number of 'b'. and We Know, that FSM or cannot be built for that and it is not regular but PDA can recongnise that ∴ We can represent in Context free Grammer

$S \longrightarrow aSb \mid \varepsilon$.

Q7(b)

True, because L_1 is a content free language and under closure properties L should be content free as well.
We Know that content free languages are closed under Kleene plus(+) and concatenation.
So, if we can make CFG for L_1 then we can also make it for L_1^+. so it means L is also content free.
Let $L_1 = \{w | w : \{a,b\}$ a in w = b in w

$S \rightarrow aSb \mid bSa \mid \varepsilon$
from it we can say $L = L1^+$.

Q8

(a) This language is not regular but content free.

CFG

$S \longrightarrow aSbb \mid \varepsilon$.

Pumping Lemma is used to prove language is ~~use to~~ regular or not and for Content free.

Let

w in a language $w = a^n b^{2n}$ for some n. According to pumping lemma theoram we can divide it into parts, $|xy| \leq n$, when we pump y which consist of 'a' the string will change but the value of x in string remain the same. Therefore there is a contradiction and it is not regular.

8

(b) $\{a^n b^m a^n b^m a^n : n, m \geq 0$

This language is not regular and content free

We can assume contradiction by using pumping lemma for regular language, consider p, such that $|s| \geq p$ can be divided into 3 parts:; for every $i \geq 0$, xy^iz is in L.

In 3 sections of 'a' have the same length and y must consist of 'b'. if we pump string once either section of 'a' not the same or 'b' so not in L.

e.g

9a

Lindenmayer or L-system is a type of formal grammer system that is used to ~~make~~ generate fractal. An L-system consist of symbols that transform each symbol into strings of symbols an initial string called axiom.
Koch curve, which is fractal starting with axiom F and applying rule F —> F + F − F − F + F for each symbol.

9b

To interpret strings as turtle program each strings is assigned a meaning that correspond to a turtle command. e.g F means move forward, + means turn left, - mean turn right. following command of symbol. This method are a way of drawing shapes using a relative cursor that moves on cartesian plane.

Q10

(a) Input □abab

result: The machine will halt immediately on ~~read~~ reading first symbol '□'.

(b)
In state R, reading '□'
machine will halt immediately and transition to halt state

c This turning machine
checks the input of 'a' and 'b'.
if this machine see ~~'b' af~~ two
consective 'b' in Mid or final
state it moves back to initial
state. On recieving 'empty' □ it
R state it will halt immedeately.