Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F29LP - Language Processors
Type of assessment:	**Individual**
Coursework Title:	CW1.1
Student Name:	Lucca Anthony Marcondes Browning
Student ID Number:	H00369673

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* Lucca Anthony Marcondes Browning

Date: *09/02/24*

Part I: Regexes & Grammars

Submission: On Canvas, under Assignments
Deadline: Week 5, Mon 12th Feb 2023
Marks possible: 28 (7% of your final mark)

Notes on answering this question:

- Regexes are enclosed within, / /, e.g. /ab+/
- Pay attention to what kind of grammar I ask for (*regular*, or *context free*, or some other type).
- For instance: if I ask for a **regular** grammar and you give me some other kind of grammar, then you may lose marks!
- In particular, if your regular grammar contains a rule of the form $S \longrightarrow abγ$ or $S \longrightarrow D$ or $S \longrightarrow \varepsilon D$, then this is not a regular grammar and your answer contains an error.
- Always clearly specify the start symbol.
- The alphabet (set of tokens) of a language cannot contain ε. ε is the empty string, that is, an absence of tokens. If your answer contains something of the form T={ε,…} (where T is supposed to be a set of tokens for your language) — then your answer is probably wrong and you're probably losing marks.

Some of you have not met set notation before, so here's a quick tutorial:
- The language determined by the regex /a*/ is { a^n | n ∈ {0,1,2,…} } or equivalently { a^n | n ≥ 0 }.
- The language determined by the regex /(a|b)?/ is { a, b, ε }.
- The language determined by the regex /a+b+/ is { $a^m b^n$ | m,n ≥ 1 } or equivalently { $a^m b^n$ | m ≥ 1, n ≥ 1 } or equivalently { $a^m b^n$ | m,n ∈ {1,2,3,…} }.
- The language determined by the English description "any nonzero digit" is { 1, 2, 3, 4, 5, 6, 7, 8, 9} or equivalently { 1, 2, … , 9} or equivalently { n | 9 ≥ n ≥ 1 } or equivalently { n | 0 < n ≤ 9 }.
- When I say "Give/Write the language …", avoid informal descriptions in English as this is often ambiguous: I expect you to use the same set theoretic notation.

1. Write a regex to identify dates of the form dd/mm/yyyy or dd.mm.yyyy, but *not* using mixed separators such as dd/mm.yyyy. *[1 mark]*

((0[1-9]|[1-2][0-9]|3[0-1])\/(0[1-9]|1[0-2])\/[0-9][0-9][0-9][0-9])|((0[1-9]|[1-2][0-9]|3[0-1])\.(0[1-9]|1[0-2])\.[0-9][0-9][0-9][0-9])

2. Write a regex for the set of even numbers without leading zeroes (base 10; so the alphabet is `[0-9]`). Note that 0 and 2 and 10 and 20 are even numbers without leading zeroes, and 00 and 02 and 1 and 01 are not even numbers without leading zeroes. Check that your regex accepts 100 and 10012. *[2 marks]*

^([02468]|[1-9][0-9]*[02468])$

3. Which of the following matches regex /`[a-zA-Z]*[^,]=`/ ? *[1 mark]*

a) `Butt=` Matches
b) `BotHEr,=` Does not Match
c) `Ample` Does not Match
d) `FIdDlE7h=` Does not Match
e) `Brittle =` Matches
f) `Other.=` Matches

4. Write the language determined by the regex /`a*b*`/ *[1 mark]*

L = {$a^p b^q$ | p≥0, q≥0}

5. Write a **regular** grammar to generate the language determined by the regex /`a*b*`/ *[2 marks]*

S ::= aS | aT | T | ε
T ::= b | bT | ε

6. Write a regular grammar to generate the language matched by /`Whiske?y`/ *[2 marks]*

S ::= WA (where "W" is terminal)
A ::= hB
B ::= iC
C ::= sD
D ::= kE
E ::= eF | F
F ::= y

7. Write a regular grammar to generate decimal numbers; the relevant regex is /`[1-9][0-9]*(\.[0-9]*[1-9])?`/.
You may find it useful to use notation resembling D ::= 0 | 1 | … | 9 to denote an set of ten production rules. *[3 marks]*

S ::= 1|2|...|9|1A|2A|...|9A|1B|2B|...|9B
A ::= 0|1|...|9|0A|1A|...|9A|0B|1B|...|9B
B ::= .C
C ::= 0C|1C|...|9C|D
D ::= 0|1|...|9

8. Give a **context free** grammar for the language $L=\{0^n1^m0^m1^n \mid n,m \in \mathbb{N}\}$ *[2 marks]*

S::= 0S1 | 0A1
A::= 1A0 | 10

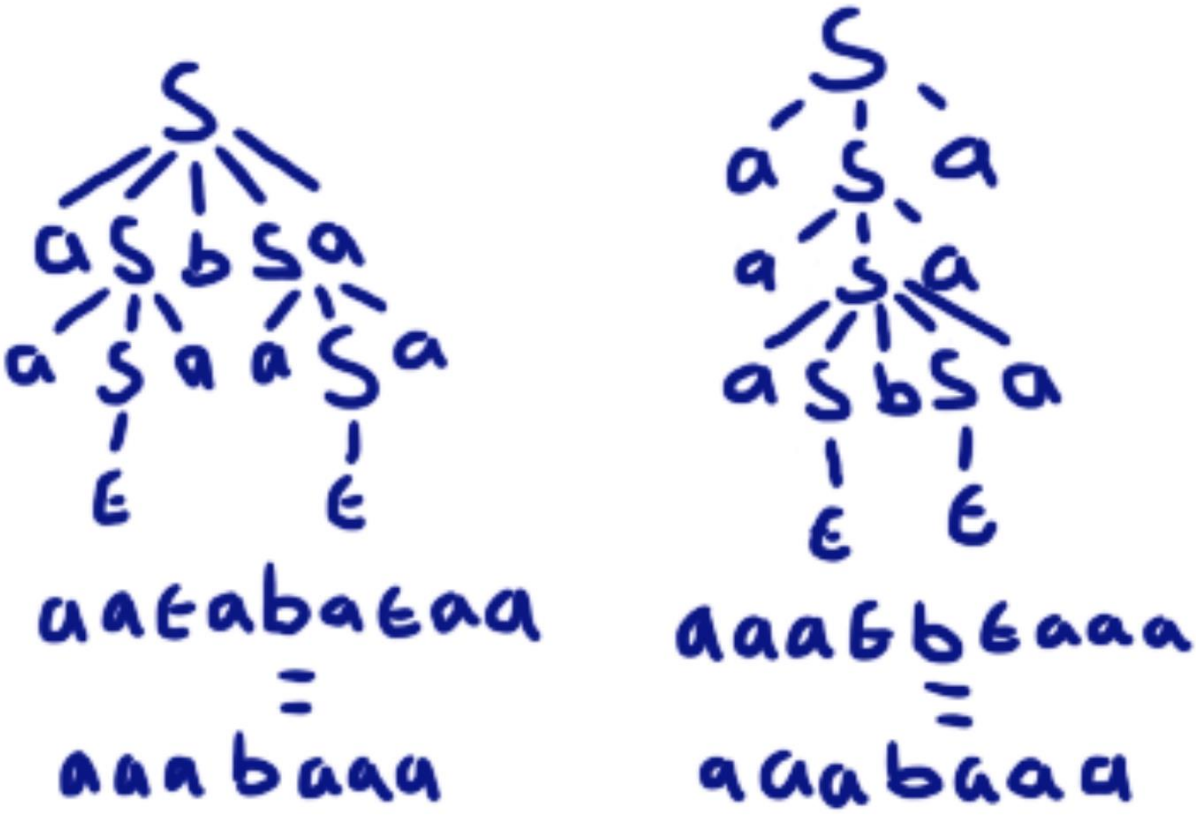
9. A parity-sequence is a sequence consisting of 0s and 1s that has an even number of ones. Give a grammar for parity-sequences. *[2 marks]*

S ::= 0S | 1A | 0
A ::= 0A | 1S | 1

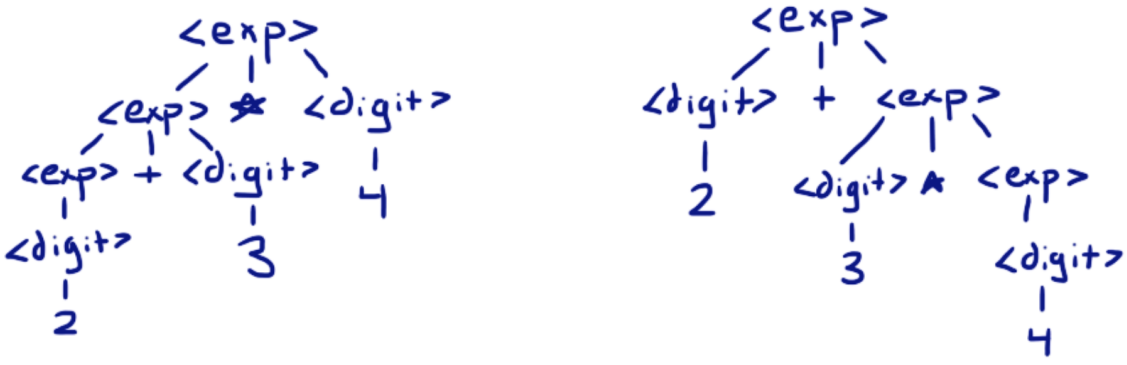
10. Construct context-free grammars that generate the following languages: *[7 marks]*

a. $(ab|ba)^*$
   S ::= ε | abS | ab | baS | ba
b. $\{(ab)^n a^n \mid n\geq 1\}$
   S ::= abAa
   A ::= abAa | ε
c. $\{w\in\{a,b\}^* \mid$ w contains exactly two bs and any number of as $\}$
   S ::= aS | bA
   A ::= aA | bB
   B ::= aB | ε
d. $\{a^n b^m \mid 0\leq n\leq m\leq 2n\}$
   S ::= A | ε
   A ::= aB | aBB
   B ::= b | Ab

11. Prove that the following grammar is ambiguous. *[3 marks]*
   a. $G=(\{S\},\{a,b\},P,S)$ with productions: $S \longrightarrow aSa \mid aSbSa \mid \varepsilon$



12. Write / Draw two distinct parse trees for $2+3*4$ and explain, in intuitive terms, the meaning of the two different parses to their denotation. *[2 marks]*



In the left tree, 2 and 3 are added together first, before being multiplied by 4, leading to the final number of 20.

In the right tree, 3 and 4 are multiplied first, before having 2 added to the product, leading to the final number of 14.

This is due to the ambiguous nature of the grammar that made $2+3*4$, and why two trees that are derived from the same expression can be entirely different and lead to entirely different products.