Examinations Winter 2021

1. Introduction

The purpose of this exam is to provide you with an opportunity to demonstrate what you have learned in this course. Because of the challenging situation imposed by COVID-19, the Faculty of Science wants to ensure that you are fully aware of the special measures we are taking to ensure the exam is fair and that you have enough time to complete and submit it.

Please read the complete cover page and sign your agreement to the academic integrity statement on page 2 before beginning your exam.

2. General information

Course (eg FCSI 101):	COMP 330		
Type of exam:	Take home exam	Time allowed:	72 hours
Date and time of exam release: (eg APR-21 09.00)	19th April 14:00		
Deadline for exam submission: (eg APR-24 09.00)	22nd April 14:00		
Method of submission:	myCourses Assignment		
Examiner:	Prakash Panangaden		
	prakash@cs.mcgill.ca		
Associate Examiner: (if relevant)	Giulia Alberini		
	giulia.alberini@mcgill.ca		

3. Terms and Conditions

a) Use of the following materials during the exam is permitted (check indicates allowed, all blank for closed book):
    course textbook................................. ☐
    your own class notes........................... ☑
    any book online.................................. ☐
    myCourses page................................. ☑
    other.. ☑   Course web site and notes there

b) Online searching of subject material/related exam questions is permitted: Yes ◯   No ◉

c) Communicating with any person regarding any aspect of the exam or course once you begin the exam **is not permitted** unless explicitly authorized.

d) Posting or sharing the exam content, including exam questions, or your answers both during and after submission **is not permitted** unless explicitly authorized.

McGill

---

4. Exam time period and submission: *Instructor please read and complete.*

a) The exam has been prepared in such a way that if undertaken under normal conditions, it can be completed in ⬜3 hours. You have been allotted ⬜72 hours to access the exam and ⬜72 hours to complete and submit the exam with your answers once you begin. Due to potential issues with internet connectivity and other unforeseen technical issues, it is strongly recommended that you plan to submit your exam script well in advance of the deadline. See also c) below.

b) If the access or completion deadline falls on a weekend day, the deadline remains.

c) For some exams, you may be allowed to submit an exam script more than once during the allotted time; only your last submission will be graded (regardless of its state of completion).

d) All exams must be submitted by the posted deadline.

e) Please check your exam script for readability and completeness before uploading and submitting.

---

5. Questions that arise during the exam

The type of questions to which an instructor can respond during an exam may be limited.

Instructor: Please describe what kinds of questions you will answer. See example given

If you find an exam question to be ambiguous or unclear, email your concerns to the instructor at the address shown above in the General Information section. Your instructor will try to review questions periodically during the exam period, but a response cannot be guaranteed. It is important, therefore, that you express any concerns over ambiguities on your exam script and still answer questions to the best of your ability

The Examiner will be responding to questions: between 9am and 5pm EDT.

---

6. Academic integrity

McGill University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/students/srr/honest/ for more information, approved by Senate on January 16, 2019).

Please affirm the following statement by either uploading a signed copy of this form; digitally signing a statement at the beginning of your submission; sending an email within myCourses; taking a photo of your signature and uploading it with the exam script; (if appropriate) responding to a 1-question quiz within myCourses that acknowledges acceptance of the integrity code.

I have neither given nor received unauthorized aid on this exam and I agree to adhere to the specific Terms and Conditions that govern this exam. If you cannot sign in the box below, just type the message in bold and type your name after it at the end of the exam.

Signature:

Belle Pan

McGill

FACULTY OF SCIENCE McGill University

FINAL EXAMINATION for Winter 2021

COMPUTER SCIENCE COMP 330

Theory of Computing

Instructions:

This exam has 5 questions. Please answer all questions. The first 4 questions are each worth 15 points and the the last question is worth 40 points. This last question consists of 10 multiple choice questions. This is an **open notes exam**, you may use the notes I posted as well as your own notes, but you are not allowed to search the internet for solutions. The questions appear starting on page 1 over 4 numbered pages. This title page and the cover pages before this one are not numbered.

Question 1 [15 points]
Classify the following language as either

1. regular or

2. context-free but not regular or

3. decidable but not context-free

You must prove your answer as explained below.

L consists of *even* length *non-empty* strings [1] over the alphabet $\{a, b\}$, so every word in L can be written as $w_1 w_2$ with $|w_1| = |w_2|$ (i.e. w_1 and w_2 have the same length); we require that w_1 starts with a and w_2 starts with b. Here are some examples of words in L: *aaabaa, abbb, aababbaa, ab*.

If you think it is regular it is sufficient to give an NFA[2]. If you think that it is context-free but not regular you have to give a PDA or CFG **and** a pumping lemma proof that it is not regular; you do not have to prove that your CFG or PDA is correct but the description should be clear. For the last case you need to give a pumping lemma proof that it is not context-free as well as an algorithm for deciding whether a given string is in the language.

Question 2 [15 points]

Suppose that I am given two regular languages R_1 and R_2 from the same fixed two-letter alphabet. Describe in outline how you would **algorithmically** test whether $R_1 \subseteq R_2$. You can assume that you are given explicit descriptions of DFA's to recognize the languages. You have to describe in *outline* the steps you could take to answer the question. The algorithm has to work without any special assumptions about the languages beyond the fact that they are regular. Your description should be *high-level*: for example, if you are going to say as some stage that you are testing whether an accept state is reachable from the start state you can just say that, you do **not** have to give the details of an algorithm to test reachability. Similarly if you want to say that you want to test something about R^* you need not give an explicit description of how you construct a DFA to recognize R^*. I am not suggesting that either of these steps is part of the required answer; they are just examples.

Question 3 [15 points]
One of the following sets is CE and the other is co-CE.

1. $\{M : |L(M)| \leq 330\}$,

2. $\{M : |L(M)| \geq 330\}$,

where M is a Turing machine description, $L(M)$ means the language recognized by M and $|L(M)|$ is the size of this language. Identify which set is CE and which is co-CE and *give proofs for both*.

---

[1] Anyone who asks whether ε is in the language will not be answered.
[2] I will not answer if you ask me whether a DFA is OK.

There is no credit for guessing, you must give the proofs. You can take it for granted that neither set is decidable.

Question 4 [15 points]
One of the following questions is decidable, the other is undecidable. For the decidable question give an outline of an algorithm to solve it. For the undecidable question give a *short* proof of undecidability by showing that it is is equivalent to a question that we know to be undecidable. Here R is a regular language and L is a context-free language specified however you prefer and the question in each case is whether the indicated inclusion holds.

- $R \subseteq L$

- $L \subseteq R$

Question 5 [40 points]
Here are 10 multiple-choice questions; each question has 4 choices. In every case, *exactly one* statement is correct. You must indicate the correct answer. You are not required to give any reasons for your answer. Each question is worth 4 points. Please just write the letter corresponding to your choice for each question. Do not write any explanations.

1. The size of the smallest DFA to recognize a regular language R

   (a) depends on the number of words in R.

   (b) depends on whether the empty string is in R.

   (c) depends on whether the language is infinite or not.

   (d) None of the above.

2. Consider the DFA shown below.
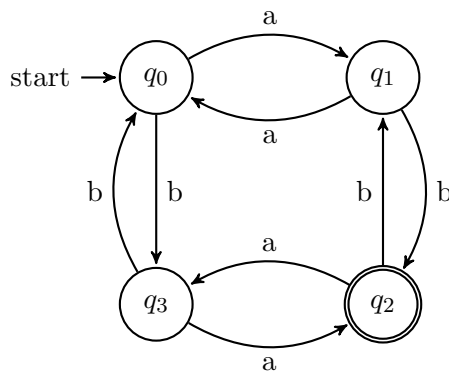
   This DFA accepts the language in which there are

   (a) an even number of a's and an odd number of b's

   (b) an odd number of a's an even number of b's

   (c) an odd number of a's and an odd number of b's

(d) an even number of a's and an even number of b's

3. The regular expression for the language accepted by the automaton above is

   (a) $((aa)^*(bb)^*)^*$.

   (b) $(abba)^*(ba^*b)^*a^*$.

   (c) it is not possible to write a regular expression for this language.

   (d) It is possible to write a regular expression but our angelic professor would never make us do such an awful problem on the final. None of the answers given in (a), (b) or (c) are correct.

4. The intersection of two deterministic context-free languages

   (a) can fail to be context free.

   (b) must be context free but not necessarily deterministic.

   (c) must be regular.

   (d) can never be context free.

5. An infinite context-free language

   (a) must contain an infinite regular subset.

   (b) need not contain any infinite regular subsets.

   (c) must never contain an infinite regular subset.

   (d) None of the above.

6. If L_1 and L_2 are languages over the same alphabet then

   (a) it is possible that $L_1 \cap L_2$ is context-free even though neither L_1 nor L_2 are context free.

   (b) if $L_1 \cap L_2$ is context-free then at least one of them must be context free.

   (c) if $L_1 \cap L_2$ is context-free then at least one of them must be regular and the other has to be context-free.

   (d) if $L_1 \cap L_2$ is context-free then both L_1 and L_2 must be context free.

7. We are interested in the question of whether two Turing machines (TM for short) accept any words in common. In symbols: given TM descriptions $\langle M_1 \rangle$ and $\langle M_2 \rangle$ we want to answer the question $L(M_1) \cap L(M_2) = \emptyset$? This question is

   (a) decidable (computable).

   (b) undecidable but computably enumerable (CE).

   (c) undecidable but **co**CE.

   (d) neither CE nor coCE.

8. Consider the set $\mathsf{FOO} = \{\langle M \rangle \,|\, |L(M)| < \infty\}$. Here we are using our standard notation, $\langle M \rangle$ means the encoding of a Turing machine. The language FOO defines

    (a) the set of Turing machine descriptions that accept only finitely many inputs

    (b) the set of Turing machine decriptions that accept *all* their inputs.

    (c) the set of Turing machine descriptions that accept infinitely many different inputs.

    (d) the set of Turing machines that accept only their own description as input.

9. The validity problem for first-order logic (FOL) is the following: given a formula ϕ of FOL we want to know if this formula is true in all interpretations. This problem is

    (a) decidable since we can always negate the formula and decide either ϕ or $\neg\phi$.

    (b) undecidable but computably enumerable since we can search through all proofs for a proof of ϕ.

    (c) not decidable or CE but it is co-CE since we can search for an interpretation that shows that it is not valid.

    (d) Neither CE nor coCE.

10. The halting problem can be solved by

    (a) asking the Internet, which has the answer to all questions.

    (b) guessing at random, your answer will be correct with probability greater than zero.

    (c) using deep learning.

    (d) There is no algorithm to solve the halting problem.