OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

SUMMER 2020 EXAMINATION

CS605

The Mathematics and Theory of Computer Science

Dr. P. Nicholl, Dr. J. Timoney, Prof. T. J. Naughton

The total time allowed is **3 hours**. This assumes an exam duration of **2 hours** plus a **1 hour** submission time. You may begin your examination up to **one hour** after the start time.

Answer **all three** questions.

By starting the examination, you confirm that you have read the instructions in the box below.

For this exam you are permitted to consult a copy of *Michael Sipser, Introduction to the Theory of Computation* (any edition) as well as any handwritten or printed notes you have compiled as part of the CS605 module. All answers must be from your own work. Collaboration is **not permitted**. The university's policies on Plagiarism will be applied. As part of the examination process you may be selected for interview after the exam period to discuss your answers. Please retain your exam answers.

This exam will be held as part of a Teams meeting. Please stay in the Teams meeting for the duration of the exam, and until Prof. Naughton has confirmed that your photos/scans of your hand-written exam answers submitted on Moodle (only PDF and JPG formats accepted) are complete and legible.

Please write the question number on top of each page. Write on one side only of each sheet of paper.

If you require any clarifications during the exam please contact Prof. Naughton either through the live Teams meeting using voice or chat, or by email to `tomn@cs.nuim.ie`. If you experience any connectivity issues that prevent you from submitting the exam **on time** you must inform the lecturer by email to `tomn@cs.nuim.ie` **within 24 hours** of the final submission time. Any emails past this time will not be considered. The email must detail clearly the nature of the issue.

If you do not have access to email you may phone Prof. Thomas J. Naughton at +353 1 7084599. This telephone number will only be available while the exam is in progress and should only be used where all other channels of communication are unavailable.

[25 marks]

1 (a) The Delvita-Praha supermarket chain has kept a record of all purchases made by each of its *L* loyalty card-carrying customers. The company offers *T* different products for sale in its supermarkets. Each customer has purchased a nonempty subset of these *T* products in their shopping history. The company wishes to offer a discount on a limited number of these *T* products, and has the following problem. Can it select only *k* different products such that each customer has at least one of the discounted products in their shopping history? You are given that this problem is in $\mathcal{NP}$. Prove that this problem is $\mathcal{NP}$-complete. You are given that SAT is $\mathcal{NP}$-complete. [13 marks]

(b) Let *L* = {<*M*> : *M* is a Turing machine with an input alphabet of {*a*,*b*} and *M* accepts at most one word, i.e. *M* either accepts no words or accepts exactly one word}. Define the complement of *L* and prove that the complement of *L* is Turing-recognisable. [12 marks]

[25 marks]

2 (a) The Delvita-Praha supermarket chain has kept a record of all purchases made by each of its *L* loyalty card-carrying customers. The company offers *T* different products for sale in its supermarkets. Each customer has purchased a subset of these *T* products in their shopping history, and each of the *T* products has been purchased by at least one customer. The company wishes to conduct a survey on the quality of its *T* different products, and has the following problem. Can it select only *k* different customers such that the union of different products purchased by each of these customers equals *T*? Prove that this problem is in $\mathcal{NP}$. [8 marks]

(b) For each of the following languages, prove that it is regular or prove that it is not regular. [10 marks]

i. {*uv* : *u*,*v* ∈ {*a*, *b*}*, *u* is not equal to *v*}
ii. {*u*#*v* : *u* ∈ {*a*}*, *v* ∈ {b}*, *u* is not equal to *v*}
iii. {*u*#*v* : *u* ∈ {*a*}*, *v* ∈ {b}*, |*u*| is not equal to |*v*|}

(c) Outline in detail how one could prove that the set of regular languages is a <u>proper subset</u> of the set of the context-free languages. You must explain each sub-proof required and give a detailed plan for how you would prove each sub-proof. The only theorems you may use (if you wish to) are those you have proved from part (b) of this question and the following [7 marks]
- a language is regular iff it is accepted by a finite automaton
- a language is context-free iff it is accepted by a pushdown automaton.

[25 marks]

3 (a) The Delvita-Praha supermarket chain has kept a record of all purchases made by each of its *L* loyalty card-carrying customers. The company offers *T* different products for sale in its supermarkets. Each customer has purchased a subset of these *T* products in their shopping history. The company has the following problem: which product has been purchased by the fewest number of different customers? Assume that there is a unique answer to this problem. Prove that this problem is in $\mathcal{P}$. [5 marks]

(b) Let the language ABC_{JAVA} be defined as ABC_{JAVA} = {<*J*, *a*, *b*, *c*> : *J* is a Java program, *a*, *b*, and *c* are integer variables declared in *J*, and throughout the execution of *J*, *a* never has the same value as *b* and *a* never has the same value as *c*}. You are given that $\text{HALT}_{\text{JAVA}}$ is undecidable. $\text{HALT}_{\text{JAVA}}$ is defined as $\text{HALT}_{\text{JAVA}}$ = {<*B*, *w*> : *B* is a Java function, and *B* halts on its string input *w*}. Prove that ABC_{JAVA} is undecidable. You may answer this question by assigning a name, mathematical construct, or piece of pseudocode to each of the numbered blanks in the proof template in Figure 1 below. Alternatively, you can choose to ignore the template and construct your own proof from scratch. [12 marks]

(c) Prove that ABC_{JAVA} is Turing-recognisable or prove that it is not Turing-recognisable. [4 marks]

(d) Give a definition of the language $\overline{\text{ABC}}_{\text{JAVA}}$ (the complement of ABC_{JAVA}). Prove that $\overline{\text{ABC}}_{\text{JAVA}}$ is Turing-recognisable or prove that it is not Turing-recognisable. [4 marks]

Proof. We will use a mapping reduction to prove the reduction ___1___. Assume that ___2___ is decidable. The function f that maps instances of ___3___ to instances of ___4___ is performed by TM F given by the following pseudocode.

F = "On input $\langle$ ___5___ $\rangle$:

1. Construct the following M' given by the following pseudocode.
   M' = "___6___"
2. Output $\langle$ ___7___ $\rangle$"

Now, $\langle$ ___7___ $\rangle$ is an element of ___8___ iff $\langle$ ___5___ $\rangle$ is an element of ___9___. So using f and the assumption that ___2___ is decidable, we can decide ___10___. A contradiction. Therefore, ___2___ is undecidable. (This also means that the complement of ___2___ is undecidable; the complement of any undecidable language is itself undecidable.)

Figure 1. Proof template that can be used if you wish. Where blanks have the same number, this denotes that their contents will be the same.