OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

SUMMER 2015 EXAMINATION

CS605

The Mathematics and Theory of Computer Science

Dr. L. Rapanotti, Dr. A. Winstanley, Mr. T. Naughton

Time allowed: 3 hours

Answer at least three questions
Your mark will be based on your best ***three*** answers

All questions carry equal marks

[25 marks]

1 Question 1 consists of thirteen multiple choice questions (MCQs) labelled (1.I) to (1.XIII). For each MCQ you will get 2 marks for choosing the correct answer, -0.25 (minus 0.25) for choosing an incorrect answer, and 0 for not choosing an answer. The maximum marks possible for Question 1 is 25 marks.

1.I Measurement of something requires a comparison. What scale do we use for algorithms?

(a) sample inputs
(b) attributes such as worst/best/average performance
(c) ensuring a correct algorithm
(d) all of the above
(e) an infinite number of algorithms requires an infinite number of scales in the worst case

1.II "All machines have equal power (in terms of computability)." This statement is

(a) provably true
(b) provably false
(c) a direct result of the Church-Turing thesis
(d) a direct result of the Invariance thesis
(e) none of the above

1.III If at least one $\mathcal{NP}$-complete problem was found to have polynomial complexity on a Turing machine then

(a) all $\mathcal{NP}$ problems would be in $\mathcal{P}$
(b) at least one $\mathcal{P}$ problem would not be in $\mathcal{NP}$
(c) the Invariance thesis could be proved
(d) all of the above
(e) none of the above

1.IV If at least one $\mathcal{NP}$-hard problem was found to have polynomial complexity on a Turing machine then

(a) all $\mathcal{NP}$ problems would be in $\mathcal{P}$
(b) at least one $\mathcal{P}$ problem would not be in $\mathcal{NP}$
(c) the Invariance thesis could be proved
(d) all of the above
(e) none of the above

1.V Any two infinite sets A and B are equinumerous (have the same cardinality) if

(a) a bijection exists between one of them and the natural numbers

(b) a bijection exists between one of them and a proper subset of the other

(c) both are finite

(d) a bijection exists between them

(e) both sets are countable (both can be enumerated or ordered)

1.VI Under what circumstances would it be preferable to use an algorithm with $(\log_2 n + 13)!$ complexity rather than an algorithm with $(2^n + 9)!$ complexity, where n is the input size?

(a) always

(b) never

(c) only when n is greater than 2

(d) only when n is greater than 3

(e) only when n is greater than 4

For the following two MCQs, consider the following sextuple of language classes: (regular, context-free, Turing-recognisable, decidable, $\mathcal{P}$, $\mathcal{NP}$).
Each language can be associated with a binary sextuple where symbol 1 denotes membership and 0 denotes nonmembership of the class in question. For example, if a language was in the first two classes and not in any of the others, it would be associated with the binary sextuple (1,1,0,0,0,0).

1.VII State the binary sextuple associated with the language of chess board configurations for which black can win.

(a) (0, 0, 0, 1, 0, 1)

(b) (0, 0, 1, 0, 0, 1)

(c) (0, 0, 1, 1, 0, 0)

(d) (0, 0, 1, 1, 0, 1)

(e) none of the above

1.VIII State the binary sextuple associated with the language of Turing machines that never go into a state 01 when executed.

(a) (0, 0, 0, 0, 0, 0)

(b) (0, 0, 1, 0, 0, 0)

(c) (0, 0, 0, 1, 0, 1)

(d) (0, 0, 1, 0, 0, 1)

(e) none of the above

1.IX Given that a k-tape deterministic Turing machine T with $k \geq 1$ can be defined by the tuple $\langle Q, \Sigma, I, q_0, F \rangle$, which of the following is always true?

(a) $|F| \geq 2$
(b) if $|F| > 2$ the TM is guaranteed to halt
(c) if $|F| < 2$ the TM might halt
(d) if $|F| = 1$ the TM is guaranteed to halt
(e) if $|F| = 1$ the TM will never halt

1.X Which of the following languages is not recursively enumerable?

(a) the set of Turing machines with an accept state labelled 99
(b) the set of nondeterministic finite automata that do not halt on the empty word
(c) the prime numbers
(d) the halting Turing machines
(e) none of the above

1.XI Which of the following languages is not recursive?

(a) the set of Turing machines with an accept state labelled 99
(b) the set of nondeterministic finite automata that do not halt on the empty word
(c) the prime numbers
(d) the halting Turing machines
(e) none of the above

1.XII The set of k-tape Turing machines is not equivalent (in terms of computational power) to which of the following sets?

(a) the set of 1-tape Turing machines
(b) the set of decidable (i.e. Turing-decidable) languages
(c) the set of partial recursive (i.e. computable) functions
(d) the set of Turing-recognisable languages
(e) the set of Java programs

1.XIII The class $\mathcal{NP}$ denotes the set of all problems

(a) that provably do **N**ot have **P**olynomial solutions
(b) for which we have **N**ot yet found **P**olynomial solutions
(c) that have **P**rovably **N**ondeterministic solutions
(d) that have **P**olynomial solutions for **N**ondeterministically-sized inputs
(e) none of the above

[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 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? Prove that this problem 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}. Prove that the complement of L is Turing-recognisable. [12 marks]

[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, 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]

4 (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 $\text{ABC}_{\text{JAVA}} = \{\langle J, a, b, c\rangle :$ *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 $\text{HALT}_{\text{JAVA}} = \{\langle B, w\rangle :$ *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.

OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

SUMMER 2015 EXAMINATION

CS605

The Mathematics and Theory of Computer Science

Dr. L. Rapanotti, Dr. A. Winstanley, Mr. T. Naughton

Declaration

To be signed by the student and collected by an invigilator at the beginning of the examination

i. I have searched through my copy of M. Sipser, *Introduction to the Theory of Computation*, any edition, (the Sipser book) and it does not contain any extra pages or annotations (except for annotations that correct minor typographical errors).

ii. I understand that by failing to notify an invigilator of any annotations or extra pages in my copies of the Sipser book, I will receive a mark of zero in this examination. This does not affect any further disciplinary actions that the University may wish to take.

iii. I understand also that directly copying large amounts of material from the Sipser books without substantially tailoring it to the question asked may result in a mark of zero.

Print name ______________________ Student number __________________

Signed ______________________________ Date ________________________