OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

SUMMER 2016 EXAMINATION

CS605

The Mathematics and Theory of Computer Science

Dr. L. Beus-Dukic, 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

<u>Additional material allowed:</u>
Copy of Michael Sipser, *Introduction to the Theory of Computation* (any edition), without annotations or extra pages. Please see declaration page at back.

[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, -0.5 (minus 0.5) for choosing an incorrect answer, and 0 for not choosing an answer. The maximum marks possible for Question 1 is 25 marks.

1.I "In order to ensure that a machine would be able to answer every question from some formal system (say, arithmetic) we would have to permit it to give incorrect answers some of the time." This statement is

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

1.II If each mathematical statement in a formal system can be either proved true or proved false then the system is

(a) consistent
(b) complete
(c) provably decidable
(d) complete or consistent but not both
(e) undefined because no sufficiently formal system can have such a property

1.III If at least one $\mathcal{NP}$-complete problem was found to have polynomial time 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 would have a proof
(d) all of the above
(e) none of the above

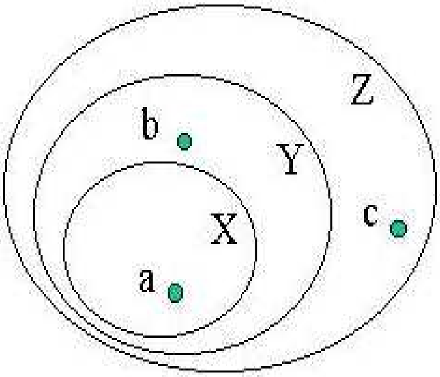
1.IV If the set of all words over alphabet Σ is countable then

(a) any language over Σ must be finite
(b) at least one language over Σ is uncountable
(c) any language over Σ is countable
(d) each language over Σ is finite
(e) none of the above

1.V The class $\mathcal{NP}$ contains (among others) problems with algorithmic solutions that

(a) can be computed in polynomial time
(b) are provably intractable
(c) are polynomial for small inputs but not for larger inputs
(d) will never terminate
(e) cannot be classified with current mathematics

1.VI Given the diagram below, depicting sets $X \subseteq Y \subseteq Z$ and elements $a \in X$, $b \in Y$, $c \in Z$, which of the following statements is false?



(a) a is X-hard
(b) b is X-complete
(c) c is Z-complete
(d) c is Y-hard
(e) none of the above

1.VII If a problem can be solved in polynomial time then
(a) the solution might still be intractable by requiring exponential space
(b) it can be verified in logarithmic time
(c) it can be verified in polynomial time
(d) it cannot be verified in all cases
(e) none of the above

1.VIII If it is proved that a solution to a problem cannot be verified in polynomial time then
(a) the solution cannot be verified in exponential time
(b) the verification process is not optimal
(c) the verification process is not decidable
(d) the problem is intractable
(e) none of the above since any solution can be verifiable in polynomial time

1.IX In the general case, by how much more powerful is a machine that allows { RR, LL, R, L, S } tape head movements than a machine that only allows { R, L } movements? Recall, 'R' denotes 'move right', 'L' denotes 'move left', 'S' denotes 'stay in this tape square', and 'RR' means move two (2) tape squares to the right in one step.
(a) The former is more powerful in terms of computability
(b) The former is more powerful in terms of asymptotic complexity but not computability
(c) The former is more powerful in terms of computability but not asymptotic complexity
(d) The former is more powerful in terms of computability and asymptotic complexity
(e) none of the above

1.X You are given Turing machine $T_1 = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$ = $\langle$ {00, 01, 50, 99}, {1}, {1, –}, δ, 00, 99, 50 $\rangle$ that operates on unary numbers, where δ is

q	***s***	***q'***	***s'***	***m***
00	1	01	1	R
00	–	99	–	L
01	1	00	1	R
01	–	99	1	S

where – is the special blank symbol, and where 'R' denotes 'move right', 'L' denotes 'move left', and 'S' denotes 'stay in this tape square'.

What does T_1 do?

(a) T_1 rounds each odd number up to the nearest even number
(b) T_1 rounds each even number up to the nearest even number
(c) T_1 increments each even number
(d) T_1 increments each number
(e) T_1 returns 1 if the number is odd

1.XI You are given Turing machine $T_2 = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$ = $\langle$ {00, 01, 02, 50, 99}, {1}, {1, –}, δ, 00, 99, 50 $\rangle$ that operates on unary numbers, where δ is

q	***s***	***q'***	***s'***	***m***
00	–	99	–	S
00	1	01	1	RR
01	1	01	1	RR
01	–	02	–	L
02	–	99	1	S
02	1	99	1	S

where – is the special blank symbol, and where 'R' denotes 'move right', 'L' denotes 'move left', 'S' denotes 'stay in this tape square', and 'RR' means move two (2) tape squares to the right in one step.

Where is the tape head positioned when T_1 halts?

(a) The tape head is positioned at the last digit on a halt
(b) The tape head is positioned to the right of the last digit on a halt
(c) The tape head is positioned at the first digit on a halt
(d) The tape head is positioned to the left of the first digit on a halt
(e) None of the above

1.XII You are given Turing machine $T_2 = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$ = $\langle$ {00, 01, 02, 50, 99}, {1}, {1, –}, δ, 00, 99, 50 $\rangle$ that operates on unary numbers, where δ is

q	***s***	***q'***	***s'***	***m***
00	–	99	–	S
00	1	01	1	RR
01	1	01	1	RR
01	–	02	–	L
02	–	99	1	S
02	1	99	1	S

where – is the special blank symbol, and where 'R' denotes 'move right', 'L' denotes 'move left', 'S' denotes 'stay in this tape square', and 'RR' means move two (2) tape squares to the right in one step.

How many table of behaviour lookups are required for T_2 as the length *n* of the input increases (where $\lceil \bullet \rceil$ denotes the ceiling operation)?

(a) $(\lceil n/2 \rceil + 1)$
(b) $(n + 1)$
(c) $(\lceil n/2 \rceil + 2)$
(d) $(n + 2)$
(e) $(n + 3)$

1.XIII The playing of certain board games pose intractable problems for machines. One such game is called GOSLOWLY. It is not necessary to understand the rules of GOSLOWLY, but you are told that even the problem of deciding which player is the winner from the state of the board is $\mathcal{NP}$-hard. However, Jill claims that this decision problem is *not* $\mathcal{NP}$-complete. Which of the following options is a true statement?

(a) If it is at least as hard as the hardest problems in $\mathcal{NP}$ then it must be in $\mathcal{NP}$, and, therefore, Jill is incorrect
(b) If it is harder than the halting problem then it must be $\mathcal{NP}$-complete
(c) If it is at least as hard as the hardest problems in $\mathcal{NP}$ but not $\mathcal{NP}$-complete then it must be outside $\mathcal{NP}$
(d) If it is $\mathcal{NP}$-hard, but not $\mathcal{NP}$-complete, then a polynomial solution may be found for it in the future
(e) If it is $\mathcal{NP}$-hard, but not actually in $\mathcal{NP}$, then a polynomial solution may be found for it in the future

[25 marks]

2 (a) A coach wishes to assemble a subset of her players to discuss how the squad performed in each match of the previous season. She has lists showing which players played in each of the *M* matches. From her full squad of *P* players, she wishes to pick a subset of *k* players such that she has at least one player from each match. Given that 3-SAT is $\mathcal{NP}$-complete, prove that this problem, called COACH, is $\mathcal{NP}$-complete. [8 marks]

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

i. $\{w = w^R : w \in \{0, 1\}^*, |w| \text{ is odd}, w\text{'s middle symbol is } 0\}$

ii. $\{w\#ww : w \in \{a, b\}^*\}$

iii. $\{w\#ww : w \in \{a\}^*\}$

(d) Consider the language $\text{CORRECT}_{FA} = \{ \langle A, M \rangle : A$ is a regular language and M is a deterministic finite automaton$\}$. Prove that CORRECT_{FA} is a decidable language. [7 marks]

[25 marks]

3 (a) 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). State the binary sextuple associated with each of the following languages.

i. The language L = {w : $w \in$ {a, b}*, |w| is odd, w has exactly one b positioned exactly in the centre of the word}. [2 marks]

ii. The language of pairs (containing a chess board configuration plus a legal move for the white player) for which that is the best next move for white. [3 marks]

iii. The language of Turing machines that go into a state 01 when executed. [3 marks]

Finally, state two (2) different binary sextuples that cannot exist, and indicate where the contradiction occurs. [2 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]

(c) How can we use a reduction to prove nonmembership of a class? [3 marks]

[25 marks]

4 (a) What does a reduction A ≤ B between two problems A and B establish about the relative computability of A and B? What does a polynomial reduction establish about the relative computational complexity of A and B? [4 marks]

(b) Let the language $\text{NTWICE}_{\text{C++}}$ be defined as $\text{NTWICE}_{\text{C++}}$ = {<*A*, *b*, *c*> : *A* is a C++ program, and *b* and *c* are integer variables declared in *A*, and throughout the execution of *A*, *b* never has a value greater than twice the value of *c*}. You are given that $\text{HALT}_{\text{C++}}$ is undecidable. $\text{HALT}_{\text{C++}}$ is defined as $\text{HALT}_{\text{C++}}$ = {<*B*, *w*> : *B* is a C++ function, and *B* halts on its string input *w*}. [13 marks]
Prove that $\text{NTWICE}_{\text{C++}}$ 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. Where blanks have the same number, this denotes that their contents will be the same. Alternatively, you can choose to ignore the template and construct your own proof from scratch.

(c) Prove that $\text{NTWICE}_{\text{C++}}$ is Turing-recognisable or prove that it is not Turing-recognisable. [4 marks]

(d) Give a definition of the language $\text{TWICE}_{\text{C++}}$ (the complement of $\text{NTWICE}_{\text{C++}}$). Prove that $\text{TWICE}_{\text{C++}}$ 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 ⟨__5__⟩ :
1. Construct the following M' given by the following pseudocode.
M' = "__6__"
2. Output ⟨__7__⟩"

Now, ⟨__7__⟩ is an element of __8__ iff ⟨__5__⟩ 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 2016 EXAMINATION

CS605

The Mathematics and Theory of Computer Science

Dr. L. Beus-Dukic, 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 ________________________