OLLSCOIL NA hÉIREANN, MÁ NUAD

NATIONAL UNIVERSITY OF IRELAND, MAYNOOTH

THIRD COMPUTER SCIENCE EXAMINATION

FOURTH COMPUTER SCIENCE EXAMINATION

THIRD COMPUTER SCIENCE AND SOFTWARE ENGINEERING EXAMINATION

FOURTH COMPUTER SCIENCE AND SOFTWARE ENGINEERING EXAMINATION

MASTERS OF COMPUTER SCIENCE (YEAR 1) EXAMINATION

MASTERS OF COMPUTER SCIENCE (YEAR 2) EXAMINATION

SAMPLE 2004

PAPER CS355/SE307/CS403

COMPUTATION AND COMPLEXITY

Mr. T. Naughton.

Attempt any THREE questions. Time Allowed: 2 hours.

1. (a) **The table of behaviour of a TM to accept L.** The start state is 00. The accept state is 99. [8 marks]

S_i	R	S_f	W	M
00	a	01	–	R
01	a	01	a	R
01	b	02	b	R
02	b	02	b	R
02	–	03	–	L
03	b	04	–	L
04	b	05	–	L
05	b	05	b	L
05	a	05	a	L
05	–	00	–	R
00	–	99	–	R

(b) **Illustration of how a reduction can be used to prove nonmembership of a class.** Given a problem x not a member of class A, by finding a reduction $x \leq y$ you would prove that y is not a member of A either. [5 marks]

(c) **Proof that W is decidable.** [8 marks]

$X =$ "On input $\langle M, w \rangle$:
1. Let $a = 2^{|w|}$.
2. Run M on w and count the number of timesteps.
3. If M halts before a timesteps, reject, otherwise accept."

X is a TM that decides W, therefore W is decidable.

(d) **Proof that PRINTERPROBLEM $\in \mathcal{NP}$.** The certificate $c = (Q_1, Q_2)$ is the two lists of jobs for the two printers. [4 marks]

$V =$ "On input $(n, P, t, (Q_1, Q_2))$:
1. Check that each element of $Q_1 \cup Q_2$ is in P : $n \times n = O(n^2)$.
2. Check that each element of P is in $Q_1 \cup Q_2$: $n \times n = O(n^2)$.
3. Check that $Q_1 \cap Q_2 = \emptyset$: $n \times n = O(n^2)$.
4. Check that the sum of $Q_1 \leq t$ and that the sum of $Q_2 \leq t$: $2n = O(n)$.
5. If all checks are passed, accept : $1 = O(1)$."

Machine V verifies PRINTERPROBLEM. V requires $O(n^2)$ timesteps in total so PRINTERPROBLEM is in $\mathcal{NP}$.

2. (a) **Definition of a model of computation.** A model of computation is a list of assumptions about the capabilities of a computing device. [5 marks]

(b) **Proof that $\text{INFINITE}_{\text{TM}}$ is undecidable.** [20 marks]

i. $\text{AT}_{\text{TM}} \leq \text{INFINITE}_{\text{TM}}$

ii. $\text{INFINITE}_{\text{TM}}$

iii. AT_{TM}

iv. $\text{INFINITE}_{\text{TM}}$

v. $\langle M, w \rangle$

vi. "On input x :
1. If $x \in \{01, 11, 100\}$, then accept x.
2. Run M on w.
3. If M accepts w, then accept x."

vii. $\langle M' \rangle$

viii. $\text{INFINITE}_{\text{TM}}$

ix. AT_{TM}

x. AT_{TM}

3. (a) **Proof that $\text{NEVEROVERFLOW}_{\text{J}}$ is undecidable.** [15 marks]

   i. $\text{A}_{\text{J}} \leq \overline{\text{NEVEROVERFLOW}_{\text{J}}}$
   ii. $\overline{\text{NEVEROVERFLOW}_{\text{J}}}$
   iii. A_{J}
   iv. $\overline{\text{NEVEROVERFLOW}_{\text{J}}}$
   v. $\langle J, w \rangle$
   vi.
```
“class Mprime {
    public static void main(String args[]) {
        int a = 0;
        if (J(w) == accept) {
            while (1 == 1) {
                a++;
            }
        }
    }
}”
```
   vii. $\langle M', a \rangle$
   viii. $\overline{\text{NEVEROVERFLOW}_{\text{J}}}$
   ix. A_{J}
   x. A_{J}

(b) **Proof that $\text{NEVEROVERFLOW}_{\text{J}}$ is not Turing recognisable.** We construct a TM M to recognise $\overline{\text{NEVEROVERFLOW}_{\text{J}}}$ as follows. [5 marks]

X = “On input $\langle J, v \rangle$:
1. Run J checking the value in v at each timestep.
2. If v overflows, accept.”

X recognises $\overline{\text{NEVEROVERFLOW}_{\text{J}}}$ therefore $\overline{\text{NEVEROVERFLOW}_{\text{J}}}$ is Turing recognisable. Since $\text{NEVEROVERFLOW}_{\text{J}}$ is undecidable, and $\overline{\text{NEVEROVERFLOW}_{\text{J}}}$ is Turing recognisable, this proves that $\text{NEVEROVERFLOW}_{\text{J}}$ is not Turing recognisable.

(c) i. $\overline{\text{NEVEROVERFLOW}_{\text{TM}}} = \{\langle J, v \rangle : J$ is a Java program, v is an integer variable declared in J, and when J is run variable v overflows at least once$\}$. [2 marks]

   ii. **Proof that $\overline{\text{NEVEROVERFLOW}_{\text{TM}}}$ is Turing recognisable.** This has been proved in 3b above. [3 marks]

4. (a) **Definition of the Church-Turing thesis.** Turing machines are equivalent to all other reasonable computing devices. [5 marks]

(b) **Proof that 2^{Σ^*} is uncountable.** Assume that 2^{Σ^*} is countable. Then it should be possible to create a list (infinite in this case) containing all of the elements of 2^{Σ^*} (all of the languages over Σ). Consider such a list of languages, and represent each language by an infinite sequence over $\{T, F\}$ where a T at the nth position indicates that the nth word in the lexicographic ordering of Σ^* is in that language, and a F at the nth position indicates that the nth word in the [10 marks] lexicographic ordering of $\Sigma*$ is not in that language. We can represent this infinite list of infinite sequences as a table, infinite in both directions. Now, if we extract the diagonal of this table, and convert each T to F and each F to T, we get a valid representation of a language over Σ that is not in the list. A contradiction, because this list was supposed to contain all such languages. Therefore our assumption was wrong and 2^{Σ^*} must be uncountable.

(c) **Placement of each language and its complement in the space of languages.** [10 marks]
The solutions will be given in the following form (smallest class the language is in, smallest class its complement is in).

i. (T-r, 2^{Σ^*})
ii. (EXP, EXP)
iii. (NP, coP)
iv. (2^{Σ^*}, 2^{Σ^*})
v. (P, P)