Dept. of Computer Science, National University of Ireland, Maynooth
CS370 – Computation and Complexity (T. Naughton)
Test 2, 6 December 2005, 17:00
Remove all paper from your desk. You have 50 minutes.
Name:
Student number:
Course:

1. Let the language $\mathrm{ADD_J}$ be defined as $\mathrm{ADD_J} = \{\langle J, a, b, c\rangle : J$ is a Java program, a, b, c are integer variables declared in J, and when J is run $c = a + b$ at least once$\}$. You are given that $\mathrm{HALT_J}$ is undecidable. $\mathrm{HALT_J}$ is defined $\mathrm{HALT_J} = \{\langle J, y\rangle : y \in \mathbb{Z}, J$ is a Java function that takes an integer argument and makes no function calls other than `System.out.Print()`, and J halts on input $y\}$.

   (a) Prove that $\mathrm{ADD_J}$ 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 their contents will be the same. Alternatively, you can choose to ignore the template and construct your own proof from scratch. [**15 marks**]

   (b) Prove that $\mathrm{ADD_J}$ is Turing recognisable or prove that it is not Turing recognisable. [**5 marks**]

   (c) Give a definition of the language $\overline{\mathrm{ADD_J}}$ (the complement of $\mathrm{ADD_J}$). Prove that $\overline{\mathrm{ADD_J}}$ is Turing recognisable or prove that it is not Turing recognisable. You may reuse your solutions to parts (b) and (c) in your proof. [**5 marks**]

2. Let the language $\mathrm{LESS_J}$ be defined as $\mathrm{LESS_J} = \{\langle J, a, b\rangle : J$ is a Java program, a, b are integer variables declared in J, and when J is run a is never less than $b\}$. You are given that $\mathrm{A_{TM}}$ is undecidable. $\mathrm{A_{TM}}$ is defined $\mathrm{A_{TM}} = \{\langle M, w\rangle : M$ is a Turing machine and M halts on input $w\}$.

   (a) Prove that $\mathrm{LESS_J}$ 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 their contents will be the same. Alternatively, you can choose to ignore the template and construct your own proof from scratch. [**15 marks**]

   (b) Prove that $\mathrm{LESS_J}$ is Turing recognisable or prove that it is not Turing recognisable. [**5 marks**]

   (c) Give a definition of the language $\overline{\mathrm{LESS_J}}$ (the complement of $\mathrm{LESS_J}$). Prove that $\overline{\mathrm{LESS_J}}$ is Turing recognisable or prove that it is not Turing recognisable. You may reuse your solutions to parts (b) and (c) in your proof. [**5 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 Turing machine 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.

3. The following Turing machine M decides the language $\{a^n b^n c^n : n \geq 0\}$. Calculate the worst-cost time and space complexity requirements of this machine as accurately as possible. As usual, assume that the *first* time each new tape cell is overwritten with a *different* value this counts as one unit of space, and assume each execution of a row of the Turing machine's table of behaviour counts as one unit of time. [**15 marks**]

M = "On input w:
1. If an X is found at the beginning of the word, jump to step 5. Otherwise, delete an a.
2. Scan to the leftmost b and replace it with X.
3. Scan to the end of the word and delete a c.
4. Go back to the beginning of the word and repeat from step 1.
5. Scan through the whole word. If there are zero or more Xs and no other symbols then accept.
6. If at any point one of these steps could not be made, then reject."