PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Complexity Theory, Semester I 2017 - IIC3242
Homework 1
Deadline: Tuesday, 21 March 2017

1 Programming Turing machines [3 points]

A *Turing machine with output* is a multi-tape Turing machine that has one specially designated write-only output tape which coincides with the last tape of the machine. If Σ is the input alphabet of the Turing machine, then a Turing machine with output is allowed to write only the symbols from $\Sigma \cup \{\mathtt{B}\}$ on the output tape. Furthermore, the content of the output tape must be of the form $w\mathtt{BBB}\cdot$, with $w \in \Sigma^*$. Each deterministic Turing machine with output which halts on all inputs computes a function $f : \Sigma^* \to \Sigma^*$, such that the value $f(w)$ is defined as the content of the output tape of this machine when ran on the input w.

The web page `https://turingmachinesimulator.com/` provides a programmable Turing machine. Here you can determine the number of tapes used, the vocabulary, etc. In this task you are asked to provide a deterministic Turing machine with output, which computes the following function $f : \Sigma^* \to \Sigma^*$:

$$f(w) = \text{ the number of 1s in } w \text{ written in binary,}$$

where $\Sigma = \{0, 1\}$. Apart from the input and output tapes you are allowed to use any number of other tapes, and any tape alphabet. Also, the content of the input tape may be rewritten as desired. However, please do notice that the output needs to be formatted properly and that the machine must halt on all inputs.

2 Computing (almost) satisfying valuations in PTIME [4 points]

We say that a function $f : \Sigma^* \to \Sigma^*$ is computable in deterministic polynomial time if there is a deterministic Turing machine with output M such that: a) M computes f; and b) there is a polynomial p such that on any input w of length n the machine M halts after at most $O(p(n))$ steps. Just as we have done in class, to show that a function is PTIME-computable, we usually give a high level description of the Turing machine computing it, or an algorithm which is easily implementable in PTIME on a TM.

Take now an alphabet $\Sigma = \{0, 1, p, \wedge, \vee, \neg\}$. We say that a function $f : \Sigma^* \to \Sigma^*$ belongs to the class $\mathcal{F}$ if it does the following on input w:

- If w represents a boolean formula φ in 3CNF, then the output of the function is an assignment v from variables of φ that satisfies at least *one half* of all the clauses in φ.
- If w does not represent a boolean formula φ in 3CNF, then the output is 0.

Notice that there are many functions that belong to the class $\mathcal{F}$. Your task is to show that there is at least one function in $\mathcal{F}$ that is computable in deterministic polynomial time. When giving the solution you are *not* required to give a Turing machine as in Problem 1, but only a high level description of the algorithm computing your function that runs in polynomial time.

Clarification: We say that w represents a formula φ in 3CNF, if each variable is of the form p_x, where x is a number in binary, and all clauses in w have precisely 3 literals. For instance $w = (p_1 \vee p_{10} \vee \neg p_{111}) \wedge (\neg p_1 \vee p_{11} \vee \neg p_{111})$ represents a formula in 3CNF, while $w = 01110 \wedge \vee$, $w = (p_1 \wedge p_{10} \vee \neg p_{111})$, or $(p_1 \vee p_{10})$ do not.

For a 3CNF formula $\varphi = (p_0 \vee p_1 \vee \neg p_{10}) \wedge (\neg p_0 \vee \neg p_1 \vee p_{10})$ the valuation $p_0 = p_1 = p_{10} = 1$ satisfies at least half of the clauses of φ.

Remark: The first step of your algorithm is allowed to be "Check that w represents a 3CNF formula". However, other steps should be spelled in greater detail.

Hint: The solution is not overly complicated. Understand what you are asked to do first (i.e. what is your input/output). Then solve it in an almost greedy fashion.