Computer Science 345
The Efficient Universe

Homework 3
Due Wednesday, March 8, 2006

No collaboration is permitted for this homework.

1 m-Reduction

Definition 1 *We say that a set of binary strings $A \subset \{0,1\}^*$ is m-reducible to the set $B \subset \{0,1\}^*$ (and denote this by $A \leq_m B$) if there exists a computable[1] function $f : \{0,1\}^* \to \{0,1\}^*$ such that for all x,*

$$x \in A \text{ if and only if } f(x) \in B.$$

Problem 1 Prove the following properties of the m-reduction.
1. m-reduction is transitive:

For all sets A, B and C if $A \leq_m B$ and $B \leq_m C$, then $A \leq_m C$.

2. If $A \leq_m B$ and A is undecidable, then B is also undecidable.

Remark: This property is the most important and useful for us.

In class we defined the halting problem and proved that it is undecidable. Recall, that the set $Halting$ is defined as follows

$$Halting = \{(P, x) : \text{the program } P \text{ halts on input } x\}.$$

In the next exercise we will reduce the $Halting$ set (problem) to several other problems, and thus show that those problems are undecidable.

Problem 2 Prove that $Halting \leq_m L$, where L is as follows.

- $L = \{P : \text{program } P \text{ halts on the empty input}\}$;
- $L = \{P : \text{program } P \text{ halts on every input}\}$.

Hint: Show that there is a computable function which on input (P, x) where P is a (description of a) program and x is a string, produces as output a (description of a) program P_x which behaves as follows: on every input P_x ignores it and instead simulates P on input x.

[1] by an algorithm which halts on every input

2 Kolmogorov Complexity

How much can we compress data, in a way that will allow us to retrieve it later? One notion is Kolmogorov Complexity.

Definition 2 *The Kolmogorov Complexity of a binary string x is the length of the shortest computer program*[2] *that prints the word x. The Kolmogorov Complexity of a word x is denoted by $K(x)$.*

Roughly speaking, the Kolmogorov Complexity of a word x is the length of the best self-extracting archive containing x. The goal of the next exercise is to show that there does not exist the best file archiver.

Problem 3 Prove the following properties of Kolmogorov Complexity.
1. There exists a positive number C s.t. for every binary word x

$$K(x) \leq |x| + C.$$

In other words, there exists a computer program of length at most $|x| + C$ that prints the word x (write this program in your favorite programming language).
2. There exist positive numbers C_1 and C_2 such that for every natural number n (below 0^n denotes a string of n zeros; n is given as a binary string):

$$K(n) - C_1 \leq K(0^n) \leq K(n) + C_1;$$
$$K(n) - C_2 \leq K(0^{2^n}) \leq K(n) + C_2.$$

Remark: In other words you need to prove upper and lower bounds on how well a string of n zeros can be compressed. Suppose now that we are given a limit (upper bound) m on the size of the file archive, and we are interested in the largest file we can compress to fit in the limit. The size of the largest file is denoted by $B(m)$. More formally, the Busy Beaver function $B(m)$ is equal to the longest string of zeros output by a program of length at most m. It turns out that $B(m)$ grows very fast.
3. Prove, that

$$B(m) \equiv \max\{n : K(0^n) \leq m\} > 2^{2^m},$$

for sufficiently large m.

Remark: Do you think this function is computable? (You do not have to answer this question in your homework.)
4. Show that for every n there exists a word x of length n with Kolmogorov Complexity at least n (i.e. $K(x) \geq n$). In other words, some strings cannot be compressed at all!

Hint: Compare the number of programs of length $< n$ and the number of strings of length n.

[2] You may think that the program is written in your favorite language: C++, C#, Java, Turing Machine or Assembly Language.

Problem 4 Now we will prove, that $K(x)$ is not computable. Suppose to the contrary that there exists a program M computing $K(x)$ (that is, $M(x) = K(x)$ for all x)[3]. Using M construct, for every n, the following program P_n:

Algorithm 1 Program P_n

Input: empty.
Output: word x.

- For each word x of length n
  - If $M(x) \geq n$ then
    return x and stop

1. Prove that the program always returns some word x and halts.
2. Prove that the Kolmogorov Complexity of the word generated by the program P_n is at least n.
3. What is the length of the program P_n? Prove that it is $O(\log n)$. Conclude that the program M does not exist.
4. Derive that determining if $K(x) \leq n$ is undecidable; or more formally, the following set is undecidable

$$S = \{(x, n) : K(x) \leq n\}.$$

3 Randomness

Problem 5 [*This is a bonus problem.*] You are given a biased coin, that when flipped, produces *Heads* with unknown probability p, where $0 < p < 1$. Show how a fair "coin flip" can be simulated (i.e. describe a random experiment that produces the events *Heads* and *Tails* with probability exactly $\frac{1}{2}$ each). You can toss the given coin multiple times.

[3]Informally, M returns the size of the best possible self-extracting archive.