# CS 1511 Homework 4

Mathew Varughese, Justin Kramer, Zach Smith

Monday, Jan 28

## 7

Assume the definition in (b) is true. Then for a recursively enumerable language L, there exists a Turing machine M with a read/write tape that is initially empty and a write-nly output tape, such that only elements of L are written to the output tape, and every element of L is eventually written to the output tape.

Now construct a turing machine $M'$.

$M'$ = "On input w:

1. Run M until it produces a new output on its output tape.
2. Check if w was the item written onto the output tape. If yes, accept.
3. Otherwise, go to step 1. If M is halted, loop indefinitely.

Machine $M'$ is the same machine defined in part (a). If $x \in L$, then $x$ will show up on the output tape of machine $M$ and will accept. If $x \notin L$, $M$ will loop indefinitely on $x$.

**8. (a)** Assume there is a function to compute the Kolmogorov complexity of a string. This function can be written into a program which has some arbitrary length x. Since this program will decide whether any string is compressible then it should compute the complexity of strings of any length, including those greater than x, or the program length. As the length of strings grows it grows constantly but the program grows by log of the length plus a constant. This means eventually the length of the string will be greater than that of the program. This raises a contradiction in the fact that the program outputted a string longer than the program itself which means that there cannot be a function to compute the Kolmogorov complexity of a string. If the complexity cannot be computed, then the set of semi-incompressible strings cannot be computed since the definition of semi-incompressible strings is strings with a complexity greater than $\sqrt{n}$. Therefore, if the complexity is not computable it cannot be proven to be more than $\sqrt{n}$.

## 8. (b)

The number of bits to represent any string with an equal amount of 0's and 1's will be $2^n$. The probability of finding an incompressible string with this property is $1/\sqrt{n}$. With this in mind, the K(x) Kolmogorov complexity will be greater than or equal to the length of the string if it's incompressible. In this case, the formula will look like this.

$n - 1/2\log_2 n + c >$ string length

As the size of n grows, eventually $1/2\log_2 n$ will grow larger than the constant c, and the equation will eventually no longer hold. Thus, there is a finite range that n can exist within where the string will be incompressible. Therefore, there are a finite amount of strings that are incompressible and have this property.

**8 (c)** Assume that the set of incompressible strings contains a infinite subset that is recursively enumerable. This means there is a mapping from the Natural Numbers this infinite subset. For example, 1 maps to $w_1$, 2 maps to $w_2$, 3 maps to $w_3$, etc. Then construct a Turing Machine M that outputs each of these strings. The subset is infinite, so $\exists x$ such that $|x| > |< M >| + a + c$. $x$ where a is the natural number that $x$ corresponds to and c is a constant. Then create $M'$ such that $M'$ outputs $x$ on input $a$. $|< M' >|$ is less than $|x|$ because $M'$ was constructed so that it is of similar length to $M$. This is a contradiction because $x$ is incompressible so no program shorter than it should be able to output it.

**8 (d)** A set is recursively enumerable if $\exists$ a Turing Machine that outputs each element in the set. Construct a turing machine as follows:

1. Start n = 0 2. Generate all programs of bit length n. 3. Run these programs all at the same time. Once one halts, If the output of that string is less than n. If not, wait til the next one halts. Output the string on to the tape. 4. Increment n and go to step 2.

This Turing Machine will run all possible programs and output strings only if they are compressible. Since it runs all programs.