Homework 8

Austin Frownfelter Matthew Bialecki

January 31, 2018

1 Problem 11

1.1 Part a

> "Show that the set of semi-incompressible strings ($K(x) \geq \sqrt{n}$) is not computable"

Assume there exists a program P_n:

```
for each string x of size n do
    If $K(x) \geq \sqrt{n}$ then output(x)
```

P_n=$\Theta(log_2(n))$ for an arbitrarily large n
Contradiction of the incompressibility of the output of P_n
Therefore the set of semi-incompressible strings is not computable.

1.2 Part b

> "Show that there are only finitely many incompressible strings that have the property that the number of bits that 0 in the string is equal to the number of bits that are 1 in the string."

Consider a program P_n which generates all strings s of length n by tossing a fair coin with 1:heads, 0:tails.
For an arbitrary n there exist at most n incompressible strings.
As $n \to \infty$, the probability $H = T \ \frac{1}{\sqrt{n}} \to 0$
Therefore, the number of incompressible strings with this property will converge to a finite number.

1.3 Part c

> "Show that the set of incompressible strings contains no infinite subset that is recursively enumerable"

Consider program P

```
for n: $0 \rightarrow \infty$
    for each string x of size n do
        If $K(x) \geq n$ then output x
```

P_n=$\Theta(log_2(n))$ for an arbitrarily large n
Contradiction of the incompressibility of the output of P_n
Since incompressible strings cannot be computed, the set of incompressible strings is not recursively enumerable.

1.4 Part D

> "Show that the set of compressible strings is recursively enumerable"

There exists a TM P:

```
for n: $1 \rightarrow \infty$
    for $(n-1)^2$ strings s of length $n$
        map s to an unused string of length $n-1$ by the mapping rule
        M
        output s
```

TM P will generate all compressible strings based on any valid mapping rule M. Therefore, P recursively enumerates the set of all compressible strings.

In other words:
For an arbitrary n, there exists a mapping to strings of length $< n$. For all strings of length n which contain a mapping to such strings, output them (enumerate them). The remainder are incompressible.
Use the above approach for all n from $1 \rightarrow \infty$. All strings which contain a mapping to a smaller string are compressible and are enumerated. The remainder are incompressible and are skipped.
Since this approach will eventually generate all strings in the language (enumerate all compressible strings), the set of compressible strings is recursively enumerable.

2 Problem 12

> "Show that for any $c > 0$ there exists strings x and y such that $K(xy) > K(x) + K(y) + c$"

There must exist a method of concatenation such that x and y can be extracted from xy. Consider a method which first encodes the length of x then x concatenated to y: xy=$|x|xy$. $|x|$ can be encoded in $log_2(|x|)$ bits. $K(xy)$ can then be rewritten as $K(x)+K(y)+log_2(|x|)$, yielding the inequality $K(x) + K(y) + log_2(|x|) > K(x) + K(y) + c$. Since for an arbitrary string, $log_2(|x|) > c$, $K(xy) > K(x) + K(y) + c$.