Course MA346H
Sample Exam Paper 1

Dr Timothy Murphy

2012–2013

1. Explain carefully what you mean by a Turing machine, and show how such a machine defines a function

$$f : \mathbb{S} \to \mathbb{S} \cup \{\perp\},$$

where $\mathbb{S}$ is the set of finite strings of 0's and 1's, and $\perp$ denotes an undefined result.

Construct two Turing machines implementing the two functions

$$[m][n] \mapsto [m+n] \text{ and } [m][n] \mapsto [mn],$$

where

$$[m] = \underbrace{1 \ldots 1}_{m \text{ 1's}} 0.$$

Answer: *A Turing machine M (following Chaitin's model) is defined by giving*

- *A finite set $Q = \{q_0, \ldots, q_{n-1}\}$ of* states*;*
- *Two maps*

$$\text{action} : \mathbb{B} \times Q \to A, \qquad \text{transition} : \mathbb{B} \times Q \to Q,$$

where $\mathbb{B} = \{0, 1\}$ and

$$A = \{\text{"}Noop\text{"}, \text{"}Swap\text{"}, \text{"}Left\text{"}, \text{"}Right\text{"}, \text{"}Read\text{"}, \text{"}Write\text{"}\}$$

is the set of 'actions'.

The machine progresses in discrete steps, which we may label with $t = 0, 1, \ldots$. At each step the machine is in a state $q(t) \in Q$.

Associated to the machine is a doubly-infinite tape T, whose configuration at any moment t is defined by a map

$$T_t : \mathbb{Z} \to \mathbb{B},$$

where $T_t(n) = 0$ for all but a finite number of n.

The machine reads from an input string

$$s = s_0 s_1 \ldots s_{m-1}$$

and writes to an output string

$$s' = s'_0 s'_1 \ldots s'_{n-1}$$

At each moment the configuration of the machine M is determined by its state $q(t)$ the configuration T_t of its tape, the portion of the input string $s_0 s_1 \ldots s_{i-1}$ which it has read in, and the output $s'_0 s'_1 \ldots s'_{j-1}$ which it has written out.

The action $a = a(t)$ taken by the machine at time t is completely determined by its state $q = q(t)$ at time t and the bit $b = T_t(0)$ (which we think of as the bit 'under the scanner' at time t), according to the map

$$\text{action} : (q, b) \mapsto a$$

The action a determines the configuration T_{t+1} of the tape at step $t+1$ as follows:

$$\begin{aligned}
a &= \text{"}Noop\text{"} : T_{t+1} = T_t \quad \textit{(ie no change)} \\
a &= \text{"}Swap\text{"} : T_{t+1}(0) = 1 - T_t(0) \\
a &= \text{"}Left\text{"} : T_{t+1}(n) = T_t(n-1) \\
a &= \text{"}Right\text{"} : T_{t+1}(n) = T_t(n+1) \\
a &= \text{"}Read\text{"} : T_{t+1}(0) = s_i \quad \textit{(read input bit)} \\
a &= \text{"}Write\text{"} : s'_j = T_t(0) \quad \textit{(write output bit)}
\end{aligned}$$

Similarly, the state $q' = q(t+1)$ of the machine at time $t+1$ is completely determined by its state $q = q(t)$ at time t and the bit $b = T_t(0)$, according to the map

$$\text{transition} : (q, b) \mapsto q'$$

Initially, at step 0, the machine is in state q_0, and the tape is blank (ie $T(n) = 0$ for all n). The machine halts if and when it enters state q_0 again.

If the machine reads in (completely) the input string s, writes out the output string s' and halts after a finite number of steps then we set

$$M(s) = s'.$$

Otherwise we set

$$M(s) = \perp.$$

First machine. *The following rules define a machine implementing the function $[m][n] \mapsto [m+n]$.*

First we read $[m]$, and output m 1's.

$$\begin{aligned}
(0,0) &\mapsto (\text{"}Noop\text{"}, 1)\\
(0,1) &\mapsto (\text{"}Read\text{"}, 2)\\
(1,1) &\mapsto (\text{"}Read\text{"}, 2)\\
(0,2) &\mapsto (\text{"}Noop\text{"}, 3)\\
(1,2) &\mapsto (\text{"}Write\text{"}, 1)
\end{aligned}$$

Now we read $[n]$ and output n 1's followed by a 0.

$$\begin{aligned}
(0,3) &\mapsto (\text{"}Read\text{"}, 4)\\
(0,4) &\mapsto (\text{"}Write\text{"}, 0)\\
(1,4) &\mapsto (3, \text{"}Write\text{"})
\end{aligned}$$

Second machine. *The following rules define a machine implementing the function $[m][n] \mapsto [mn]$.*

First we read $[m]$, and store it on the tape.

$$\begin{aligned}
(0,0) &\mapsto (\text{"}Noop\text{"}, 1)\\
(0,1) &\mapsto (\text{"}Read\text{"}, 2)\\
(1,1) &\mapsto (\text{"}Read\text{"}, 2)\\
(0,2) &\mapsto (\text{"}Left\text{"}, 3)\\
(1,2) &\mapsto (\text{"}Right\text{"}, 1)
\end{aligned}$$

Now we read a bit, and if it is 0 we output 0 and halt.

$$\begin{aligned}
(0,3) &\mapsto (\text{"}Read\text{"}, 4)\\
(0,4) &\mapsto (\text{"}Write\text{"}, 0)
\end{aligned}$$

If it is 1 we output m 1's.

$$(1,4) \mapsto (\text{"}Noop\text{"}, 5)$$
$$(0,5) \mapsto (\text{"}Noop\text{"}, 7)$$
$$(1,5) \mapsto (\text{"}Write\text{"}, 6)$$
$$(1,6) \mapsto (\text{"}Left\text{"}, 5)$$

Now we return to the right-hand end of the tape

$$(1,7) \mapsto (\text{"}Right\text{"}, 7)$$

... and read another bit

$$(0,7) \mapsto (\text{"}Left\text{"}, 3)$$

2. Given sets X, Y, what is meant by saying that

   (a) $\#X = \#Y$,

   (b) $\#X \le \#Y$?

   Show that

$$\#X \le \#Y \text{ and } \#Y \le \#Y \implies \#X = \#Y.$$

Answer:

(i) **(a)** $\#X = \#Y$ *means 'there exists a bijection* $f : X \to Y$*'.*

(b) $\#X \le \#Y$ *means 'there exists a injection* $f : X \to Y$*'.*

(ii) *By hypothesis there exist injections*

$$f : X \to Y, \; g : Y \to X.$$

We have to construct a bijection

$$h : X \to Y.$$

For simplicity, we assume that X *and* Y *are disjoint (taking disjoint copies if necessary).*
Given $x_0 \in X$*, we construct the sequence*

$$y_0 = f(x_0) \in Y, \; x_1 = g(y_0) \in X, \; y_1 = f(x_1) \in Y, \ldots.$$

There are two possibilities:

(i) *The sequence continues indefinitely, giving a singly-infinite chain in X:*

$$x_0, y_0, x_1, y_1, x_2, \ldots$$

(ii) *There is a repetition, say*

$$x_r = x_s$$

for some $r < s$. Since f and g are injective, it follows that the first repetition must be

$$x_0 = x_r,$$

so that we have a loop

$$x_0, y_0, x_1, y_1, \ldots, x_r, y_r, x_0.$$

In case (i), we may be able to extend the chain backwards, if $x_0 \in \mathrm{im}(g)$. In that case we set

$$x_0 = gy_{-1},$$

where y_{-1} is unique since g is injective.
Then we may be able to go further back:

$$y_{-1} = fx_{-1},\ x_{-2} = gy_{-1},\ \ldots.$$

There are three possibilities:

(A) *The process continues indefinitely, giving a doubly-infinite chain*

$$\ldots, x_{-n}, y_{-n}, x_{-n+1}, y_{-n+1}, \ldots, x_0, y_0, x_1, \ldots.$$

(B) *The process ends at an element of X, giving a singly-infinite chain*

$$x_{-n}, y_{-n}, x_{-n+1}, \ldots.$$

(C) *The process ends at an element of Y, giving a singly-infinite chain*

$$y_{-n}, x_{-n+1}, y_{-n+1}, \ldots.$$

It is easy to see that these chains and loops are disjoint, partitioning the union $X + Y$ into disjoint sets. This allows us to define the map h on each chain and loop separately. Thus in the case of *a doubly-infinite chain or a chain starting at an element* $x_{-n} \in X$, *or a loop, we set*

$$hx_r = y_r;$$

while in the case of a chain starting at an element $y_{-n} \in Y$ *we set*

$$hx_r = y_{r-1}.$$

Putting these maps together gives a bijective map

$$h : X \to Y.$$

3. Define the *algorithmic entropy* $H(s)$ of a string s (of 0's and 1's).

   Show that

$$H(s) \leq |s| + H(|s|) + O(1).$$

   Show conversely that there exists a constant C such that for each $n \in \mathbb{N}$ there exists a string s of length n such that

$$H(s) \geq n + H(n) - C.$$

Answer: *Suppose* T *is a Turing machine. We set*

$$H_T(s) = \min_{p:T(p)=s} |s|;$$

and we set

$$H(s) = H_U(s),$$

where U *is a universal machine, chosen once and for all.*

In other words, $H(s)$ *is the length of the shortest string* p *which when input into* U *will output* s.

Suppose we chose another universal machine V *in place of* U. *By the definition of a universal machine, there exist strings* u, v *such that*

$$U(vs) = V(s), \quad V(us) = U(s).$$

It follows that

$$H_U(s) \leq H_V(s) + |v|, \quad H_V(s) \leq H_U(s) + |u|.$$

Thus

$$H_V(s) = H_U(s) + O(1).$$

Now suppose $|s| = n$. We have to show that

$$H(s) \leq |s| + H(n) + O(1),$$

where $H(n) = H(\langle n \rangle)$, the binary encoding for n.

Let ν be a string of minimal length such that

$$U(\nu) = B(n).$$

We construct a machine T which outputs s on input νs.

T starts by imitating U. Thus it reads in ν and computes $B(n)$. However, instead of outputting $B(n)$ it saves it; and instead of halting after this computation it passes to the next stage, and which it reads in and outputs n bits, ie the whole of s.

Thus

$$T(\nu s) = s$$

and so

$$U(\langle T \rangle \nu s) = s.$$

It follows that

$$\begin{aligned} H(s) &\leq |\langle T \rangle| + |\nu| + |s| \\ &\leq O(1) + H(n) + |s|, \end{aligned}$$

as required.

To prove the converse, we use the result that

$$H(s,t) = H(s) + H(t|s) + O(1),$$

where $H(s,t)$ is the joint entropy of s and t, and $H(s|t)$ is the conditional entropy of s given t.

We shall show that

(a) *For all $s \in \mathbb{S}$,*

$$H(s, B(|s|)) = H(s) + O(1),$$

(b) *For each $t \in \mathbb{S}$ we can find a string s of length n such that*

$$H(s|t) \geq n.$$

The result will follow on taking $t = B(n)$ in (b). For then

$$\begin{aligned} H(s) &= H(s, B(n)) + O(1) \\ &= H(B(n), s) + O(1) \\ &= H(n) + H(s|B(n)) + O(1) \\ &\geq H(n) + n + O(1). \end{aligned}$$

To prove (a) it is sufficient to observe that we can modify U to construct a machine which outputs $\langle s \rangle \langle |s| \rangle$ when U outputs s.

For (b) recall the definition of $H(s|t)$. Let $\mu = \mu(t)$ be the shortest input for U to output t. Then

$$\begin{aligned} H(s|t) &= \min_{T,p:T(\mu p)=s} |\langle T \rangle| + |p| \\ &= \min_{T,p:U(\langle T \rangle \mu p)=s} |\langle T \rangle| + |p| \end{aligned}$$

Consider t fixed. Then to each $s \in \mathbb{S}$ we can associate the string $\langle T \rangle p$ of length $H(s|t)$. It is a straightforward matter to verify that the map

$$s \mapsto \langle T \rangle p$$

is injective.

Now consider the 2^n strings s of length n. Thus at least one s must map to a string of length $\geq n$, since there are only $2^n - 1$ strings of length $< n$. It follows that at least one s must have

$$H(s|t) \geq n,$$

as claimed.