1-tape Turing Machines

This talk is about bitstrings and machines. A bitstring is a sequence of zeroes and ones like

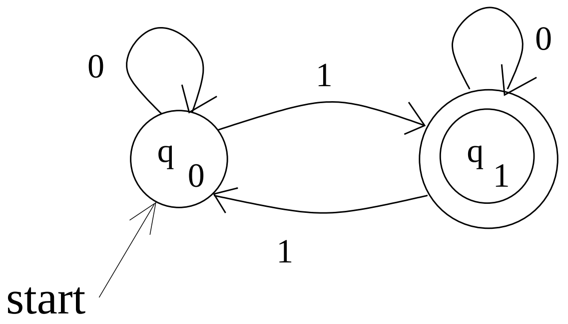
$$1010010$$

or

$$10101$$

The machines, which are imaginary, simplified, computers, are able to decide which bitstrings have a certain property. The weakest machine is a FINITE AUTOMATON.

A finite automaton has STATES. Its 'INPUT STRING' is written on a TAPE which it reads strictly from left to right, altering its STATE according to a TRANSITION FUNCTION δ. It 'accepts' the string if it ends up in an ACCEPTING state.



δ	0	1
q_0	q_0	q_1
q_1	q_1	q_0

accepting states: $\{q_1\}$

$$q_0\ \boxed{1}\ q_1\ \boxed{0}\ q_1\ \boxed{1}\ q_0\ \boxed{0}\ q_0\ \boxed{0}\ q_0\ \boxed{1}\ q_1\ \boxed{0}\ q_1$$

The most powerful machine is a TURING MACHINE. For example, there is a Turing machine capable of proving every theorem of mathematics.

A Turing machine has STATES and a TAPE and TRANSITIONS like a finite automaton, but it can MOVE both LEFT and RIGHT and it can WRITE on the tape. Its actions are specified by QUINTUPLES. (Note: B means BLANK.)

The following machine looks for palindromes (ABLE WAS I ERE I SAW ELBA).

$q_0\ 0\ B\ R\ q_1$	$q_0\ 0\ B\ R\ q_3$
$q_1\ 0\ 0\ R\ q_1$	$q_3\ 0\ 0\ R\ q_3$
$q_1\ 1\ 1\ R\ q_1$	$q_3\ 1\ 1\ R\ q_3$
$q_1\ B\ B\ L\ q_2$	$q_3\ B\ B\ L\ q_4$
$q_2\ 0\ B\ L\ q_5$	$q_4\ 1\ B\ L\ q_5$
$q_2\ 1\ 1\ R\ q_6$	$q_4\ 0\ 0\ R\ q_6$

In state q_0 the machine is reading the leftmost nonblank symbol on the string.

If it is a 0, the machine scans to the right in state q_1, and inspects the rightmost nonnlank symbol: if a 0, it enters state q_5 and moves to the leftmost nonblank symbol; if a 1 it enters state q_6 and stops.

If it is a 1, the machine looks for a 1 at the right-hand end.

$q_5\ 0\ 0\ L\ q_5$
$q_5\ 1\ 1\ L\ q_5$
$q_5\ B\ B\ R\ q_0$

The accepting states are q_0, q_2, and q_4.

TURING MACHINES are MUCH more powerful than finite automata. However, we are going to prove a remarkable theorem due to Hartmanis ('On the complexity of one tape Turing machine computations,' JACM 15 (1968), 325–339.)

DEFINITION:

$$T_M(n)$$

is the maximum number of steps taken by a Turing machine M on length-n bitstrings.

If M is a Turing machine or a finite automaton, then

$$L(M)$$

is the set of strings accepted by M.

A set of strings is a REGULAR SET if they are accepted by some finite automaton.

FOR EXAMPLE, the set of finite-precision decimal numbers, such as 3.14159, is a regular set.

Palindromes are not a regular set. The 'palindromes' machine satisfies

$$T_M(n) = (n+1)(n+2)/2 - 1$$

whereas ... here is the *remarkable* theorem mentioned above.

THEOREM (Hartmanis): Let M be a Turing machine. If

$$\lim_{n\to\infty} \frac{T_M(n)}{n \log n} = 0,$$

then $L(M)$ is a regular set.

The palindromes example give examples of *crossing sequences* associated with the computation.

DEFINITION: A *crossing sequence* is the pile of states occurring when crossing (back-and-forth) between two adjacent tape squares.

DEFINITION: $H_M(n)$ is the maximum height of crossing sequences on inputs of length n.

There are two other properties of crossing sequences.

> LEAP OF FAITH: if $H_M(n)$ is BOUNDED then $L(M)$ is a regular set, that is, M is no more powerful than a finite automaton.

The SPLICING PRINCIPLE below is illustrated by a rather weak example:

$q_0 0 x R q_2$	$q_2 1 1 R q_1$
$q_0 1 1 R q_1$	$q_1 B B L q_3$
$q_1 0 x R q_1$	$q_3 x B L q_3$
$q_1 1 1 R q_2$	$q_3 1 B L q_3$
$q_2 0 x R q_2$	Accepting: $\{q_3\}$

SPLICING PRINCIPLE: If in a computation, the crossing sequences at two points i and j are identical, then one can shorten the input string by erasing the symbols between i and j, and erase the crossing sequences from $i+1$ to j, and the resulting list of crossing sequences is compatible with the computation on the shorter string.

FIRST THEOREM: if $L(M)$ is not a regular set, so $H_M(n)$ is unbounded, then $\lim_n H_M(n)/\log n > 0$.

Proof. Given any $h > 0$ there is an n such that $H_M(n) \geq h$. Let x be a SHORTEST input string achieving a height $\geq h$. (It does not matter whether or not M accepts x).

Let $c_0, c_1, \ldots c_n$ be the crossing sequences derived from the computation on x.

Claim that no three of these crossing sequences c_i are the same.

Suppose c_i, c_j, and c_k are the same. By the splicing principle we can remove the symbols between i and j or between j and k. If between c_i and c_j all the crossing sequences have height $< h$, splice from i to j, otherwise splice from j to k.

In this way we get a shorter string with the SAME maximum height, contradicting the choice of x. Hence every crossing sequence occurs at most twice. Let n be the length of x.

Let k be the number of states of M. Then

$$\boxed{\text{For any } r\text{, there are } k^r \text{ possible crossing sequences of height } r.}$$

$$2\sum_{r=0}^{h} k^r \geq n+1$$

$$\frac{k^{h+1}-1}{k-1} \geq \frac{n+1}{2}, \quad k^{h+1} > \frac{k-1}{2}(n+1)$$

so (assuming without loss of generality $k \geq 3$)

$$h+1 \geq \log_k(n+1)$$

But $h \leq H_M(n)$, so $H_M(n)$ grows at least as rapidly as $\log n$. Q.E.D.

Now to prove the main theorem: if

$$\lim_{n\to\infty} \frac{T_M(n)}{n\log n} = 0,$$

then $L(M)$ is a regular set.

Equivalently, if $L(M)$ is not a regular set, then

$$\lim_{n\to\infty} \frac{T_M(n)}{n\log n} > 0.$$

Proof. Assuming that $L(M)$ is not a regular set, $H_M(n)$ is unbounded. We again consider the string x of the previous theorem: no crossing sequence occurs more than twice.

Let

$$c_0, c_1, \ldots, c_n$$

be the crossing sequences from the computation, but this time they are arranged not left-to-right, but in nondecreasing order of height. Write down their heights as

$$h_0, h_1, \ldots h_n$$

in nondecreasing order.

Let $m_0, m_1, \ldots,$ be the list

$$0, 0, 1, \ldots, 1, 2, \ldots, 2, 3, \ldots$$

where 0 occurs twice, 1 occurs $2k$ times, 2 occurs $2k^2$ times, and so on. Because no crossing sequence occurs more than twice, for each r, $h_i = r$ for at most $2k^r$ subscripts i: the h_i-list can be got by *removing* entries from the m_i-list. It follows that

$$h_i \geq m_i \quad \text{for all } i.$$

It is easy to show that

FACT: Number of steps in a computation = sum of heights of crossing sequences.

so $T_M(n) \geq \sum h_i \geq \sum_{i=0}^{n} m_i$.

Let $h = \log_k(n+1) - 1$. In the previous theorem it was shown that $m_n \geq h$; also, $k^{h+1} > (k-1)(n+1)/2$.

Since $m_n \geq h$, the sequence m_i $(0 \leq i \leq n)$ includes all $2k^{h-1}$ occurrences of $h-1$, so (we replace $n+1$ by n for simplicity):

$$T_M(n) \geq \sum_i m_i \geq 2k^{h-1}(h-1) \geq \frac{2}{k^2} k^{h+1}(h-1)$$
$$> \left(\frac{2}{k^2}\frac{k-1}{2}\right) \; n \; (\log_k(n) - 2),$$

and $\lim_{n\to\infty} T_M(n)/(n \log n) > 0$. Q.E.D.