Turing Machines

A **Turing Machine** M is a 5-tuple

$M = (Q, X, q_0, F, \delta)$ where

Q set of states

X tape alphabet, $X' = X \cup \{B\}$, B = Blank

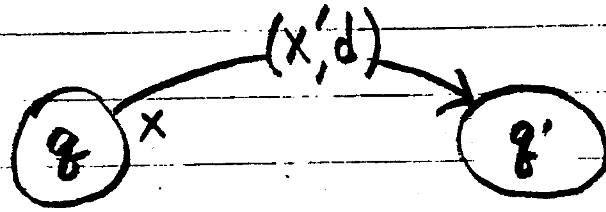
q_0 start symbol, F accepting states

$\delta : Q \times X' \longrightarrow Q \times X' \times \{L, N, R\}$

represent δ by a list of 5-tuples:

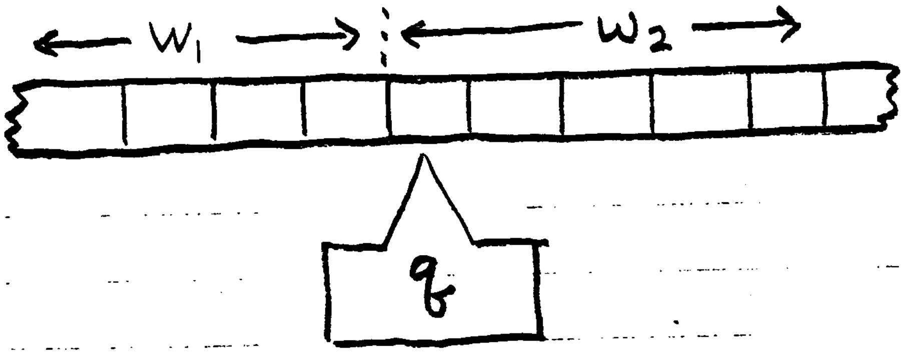
(q, x, q', x', D) iff $\delta(q, x) = (q', x', D)$.

or by a directed graph



Instantaneous Descriptions (ID's)

$w_1 \, q \, w_2$



$w_1 \, q \, w_2$ is equivalent to

$B w_1 \, q \, w_2$ and to $w_1 q w_2 B$.

Assume that ID has form

$$\overline{w_1} \, y \, q \, x \, \overline{w_2}$$

(1) $\delta(q, x) = (q', x', L)$

new ID $= \overline{w_1} \, q' \, y \, x' \, \overline{w_2}$

(2) $\delta(q, x) = (q', x', R)$

new ID $= \overline{w_1} \, y \, x' \, q' \, \overline{w_2}$

(3) $\delta(q, x) = (q', x', N)$

new ID $= \overline{w_1} \, y \, q' \, x' \, \overline{w_2}$

M halts on the ID $w_1 q w_2$ if neither $w_1 q w_2$ nor $w_1 q w_2 B$ has the form $w_1 q x \overline{w_2}$ for a pair (q, x) on which $\delta(q, x)$ is defined.

If α and α' are two ID's, then we write $\alpha \Rightarrow \alpha'$ if M can directly transform α to α' by one of the three rules on the preceding page. $\overset{*}{\Rightarrow}$ will be the reflexive, transitive closure of $\Rightarrow$.

The language accepted by a Turing machine M over alphabet X is the set

$$T(M) = \{ w \in X^* \mid \exists\, w_1, w_2 \in (X')^* \; [\, q_0 w \overset{*}{\Longrightarrow} w_1 q_a w_2 \text{ where } q_a \in F \text{ and } w_1 q_a w_2 \text{ is a halting ID} \,] \}$$

example ①

Language of balanced parenthesis

```
( ( ( ) ) ( ) )
( ( ( # ) ( ) )
( ( # # ) ( ) )
( ( # # # ( ) )
( # # # # ( ) )
( # # # # ( # )
( # # # # # # )
( # # # # # # #
# # # # # # # #
              ↑
```

$\langle q_0, \text{`('}, q_0, \text{`('}, R \rangle$

$\langle q_0, \#, q_0, \#, R \rangle$

$\langle q_0, \text{`)'}, q_1, \#, L \rangle$

$\langle q_0, B, q_2, B, L \rangle$

$\langle q_1, \text{`('}, q_0, \#, N \rangle$

$\langle q_1, \#, q_1, \#, L \rangle$

$\langle q_2, \#, q_2, B, L \rangle$

$\langle q_2, B, \text{Halt}, B, N \rangle$

Can also think of Turing Machines as devices for computing functions.

Suppose that ϕ is a function with k arguments and that $\phi(n_1, \ldots, n_k) = m$, then the initial ID will be

$$q_0\, 1^{n_1} B\, 1^{n_2} B \ldots 1^{n_k} B$$

If M computes ϕ, then the final Id will be

$$q_h\, 1^{m} B$$

where $q_h \in F$ is a halting state.

<u>Example</u> (2) unary subtraction

$$\phi(m, n) = \begin{cases} n - m & n \geq m \\ ??? & \text{o.w.} \end{cases}$$

initially: $B\,1^{m} B\, 1^{n} B$

$n-m$

finally: $B\, 1^{m-n}\, B$

$(q_0, 1, q_1, B, R)$

(q_0, B, q_0, B, N) (q_0 is an accepting, halting state.)

$(q_1, 1, q_1, 1, R)$

(q_1, B, q_2, B, R)

$(q_2, 1, q_2, 1, R)$

(q_2, B, q_3, B, L)

$(q_3, 1, q_4, B, L)$

(q_3, B, q_0, B, N)

$(q_4, 1, q_4, 1, L)$

(q_4, B, q_5, B, L)

$(q_5, 1, q_5, 1, L)$

(q_5, B, q_0, B, R)

Busy Beaver Function

Suppose that M_n is a Turing Machine over $X = \{1\}$ with n states $q_0, \ldots, q_{n-1}$. Start the machine on a blank tape in state q_0. Either the machine will halt or it will run forever. If it does stop, count the number of 1's on its tape. Let $P(n)$ be the maximum number of 1's that can be produced as output by any n state machine if and when it halts. Here is a three state machine that prints 1^6. Thus $P(3) \geqslant 6$.

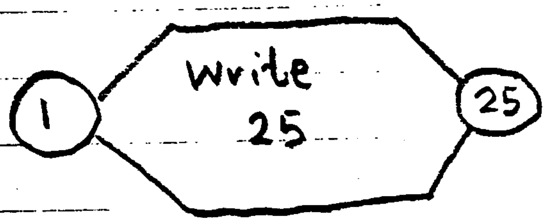
1. $(q_1, B, q_2, 1, R)$ $(q_1, 1, \text{halt}, 1, N)$
2. (q_2, B, q_3, B, R) $(q_2, 1, q_2, 1, R)$
3. $(q_3, B, q_3, 1, L)$ $(q_3, 1, q_1, 1, L)$

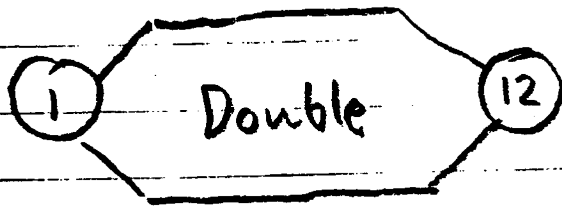
Is $P(n)$ Turing machine computable?

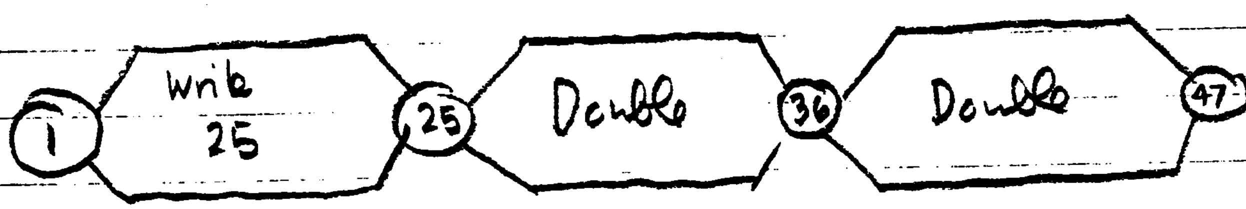
Some observations about P(n)

1) $P(1) = 1$ easy

2) $P(47) \geq 100$

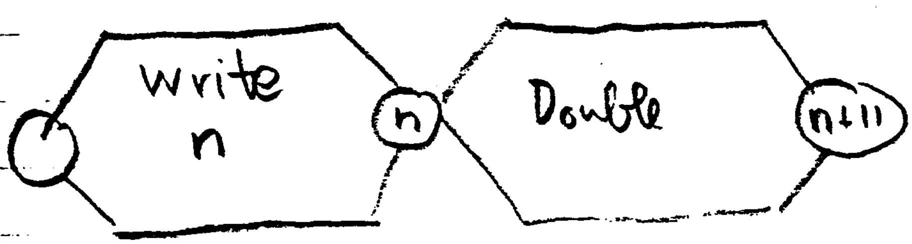






2) $P(n+1) > P(n)$ easy

3) $P(n+11) \geq 2n$



Proof that the Function $P(n)$ is not Turing computable:

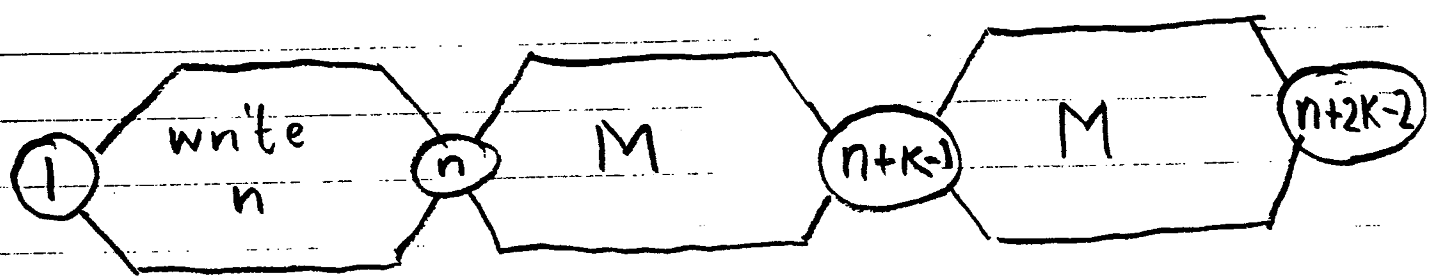
Assume that $P(n)$ is computable by some machine M with K states

$$q_0 1^n B$$

run M

$$q_a 1^{P(n)} B$$

Hence, there must be a machine with $n+2K-2$ states that will print $P(P(n))$ 1's.



Since $p(n+1) > p(n)$, we have

$p(i) > p(j)$ if $i > j$.

Thus, by contraposition

$j \geq i$ if $p(j) \geq p(i)$.

$p(n+2K-2) \geq p(p(n))$

So, $n+2K-2 \geq p(n)$.

Since n is arbitrary, we have

$$(n+11)+2K-2 \geq p(n+11)$$

$$n+11+2K-2 \geq 2n$$

$$9+2K \geq n$$

But n is arbitrary and K is fixed, so this is a contradiction.