① Consider ordered pairs (v, w) where w is an integer written in binary and $v = 3w$ is written in binary. Draw a diagram of a TM w/c accepts only such ordered pairs.

Assumptions: ① The tape alphabet Γ is $\{X, \text{'('}, \text{','}, \text{')'}, 0, 1, \square\}$ ② The input alphabet is $\Sigma = \{\text{'('}, \text{','}, \text{')'}, 0, 1\}$. ③ Integers are written in little endian fashion: i.e., the number 2 is written as 01.

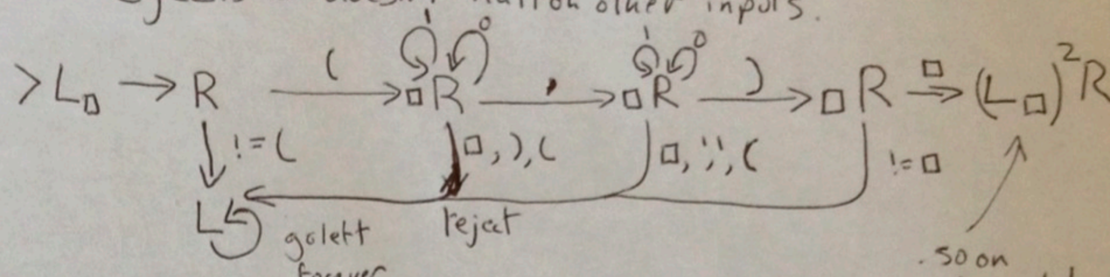
Denote by R_a (L_a) the machine that scans right (left) till it sees the first symbol a.

$>R \circlearrowleft^{!=a}$, $>L \circlearrowright^{!=a}$

to be continued

$\Rightarrow$

The first machine we build checks the input has the format (binary number, binary number). It outputs a string binary number□binary number on okay inputs and rejects or doesn't halt on other inputs.



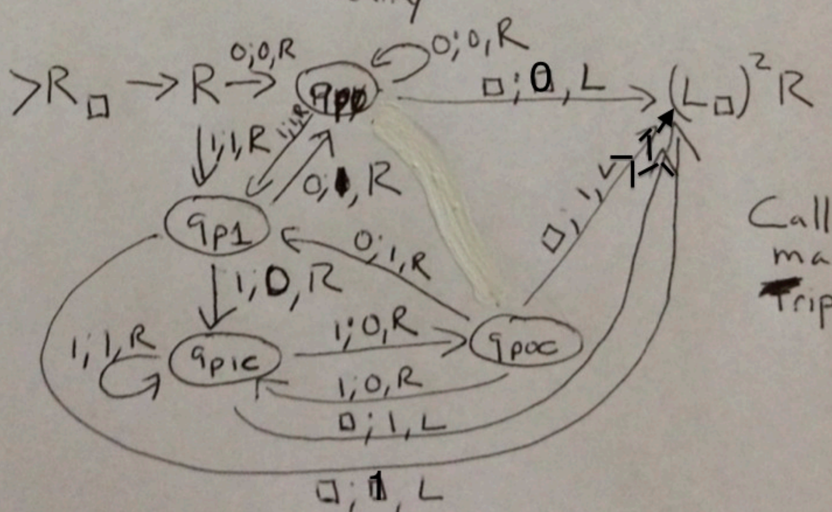
Call this machine Check-Input.

We next build a machine w/c on input:

~~binary#□binary#~~ v□w output v□3w
(written in binary)

Let prev0 mean previous digit was a 0. We further abbreviate as pø
prev1 mean previous digit was a 1 We " " as p1
Let carry denote carry bit set. We " " as c

We make use of states qpøc, to say previous digit ø carry bit set, qp1c; qpø; qp1.
(no carry)

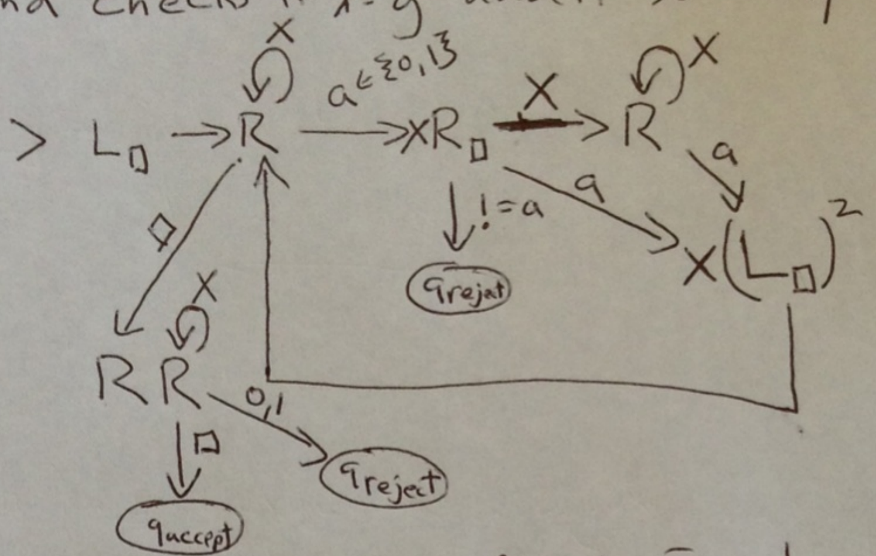


Call this machine Triple

Our last machine takes input of the form

$$\underbrace{x}_{\text{binary}} \sqcup \underbrace{y}_{\text{binary}}$$

and checks if $x = y$ and if so accepts, else rejects.



Call this machine Check-Equal.

So the whole machine for problem 1 is:

> Check-Input → Triple → Check-Equal

3 Show if L is recognized by a usual TM it is recognized by a hyperspace TM.

<u>Proof</u> Since a usual TM can be simulated by a TM with a semi-infinite tape, Let's assume L is recognized by some machine M that uses a semi-infinite tape. We will simulate it with a hyperspace machine HM.

The tape alphabet of HM will be that of M together with symbols $\underline{b}$ for each symbol b of M's alphabet.

The transition f of HM on transitions that ~~move left.~~ For a transition of M that moves right, for example,

do same as M.

$$\delta(q, a) = (q', b, R)$$

HM will do the following

$$\delta'(q, a) = (q_{a1}, \underline{b}, H)$$

hyperspace

we are hoping this right advances us two squares. If not, we detect and reject

$$\delta'(q_{a1}, a') = (q_{a2}, \underline{a'}, L)$$

a' != underscore char

for underscore char reject

mark square jump to

$$\delta'(q_{a2}, a'') = (q_{a3}, a'', L)$$

a'' != underscore char

For underscore char reject

$$\delta'(q_{a3}, \underline{b}) = (q_{a4}, b, H)$$

for any char != $\underline{b}$ reject

unmark starting place hope hyper jump advances us two again

$$\delta'(q_{a5}, \underline{a'''}) = (q', a''', L)$$

must be underscore char or reject.

If the hyperspace machine didn't reject and completed this sequence, it will be one to the right of where it was when it read a. Further, it will have written to, and left all other squares unchanged. So our simulation works.