Login/CID: hg1523/02410004 name: Honglei Gu

(please forgive me for using markdown to write the solution, I know you consider it a waste of time, I dont want to cause any confusion with the photography and possibly drawing, the graph are drawn with iodraw, I am very familiar with the tools and please forgive me for using these)

1 Exercise 1

Sepecify a Turning Machine (TM) $M = (Q, \sum, s, \delta)$ which starts a tape with the representation of a list $L = [x_1, x_2, \dots, x_n]$(as in lectures) and terminates with a tape representing the (singleton) list [s] where

$s = n + \sum_{i=1}^{n} x_i$

Describe the computational steps (configuration) of M when the initial tape represents the list [1,2,3]

the tape starts with [1,2,3] or $01B1^2B1^30$

should end with [9] or 01^90

let B represent the blank symbol

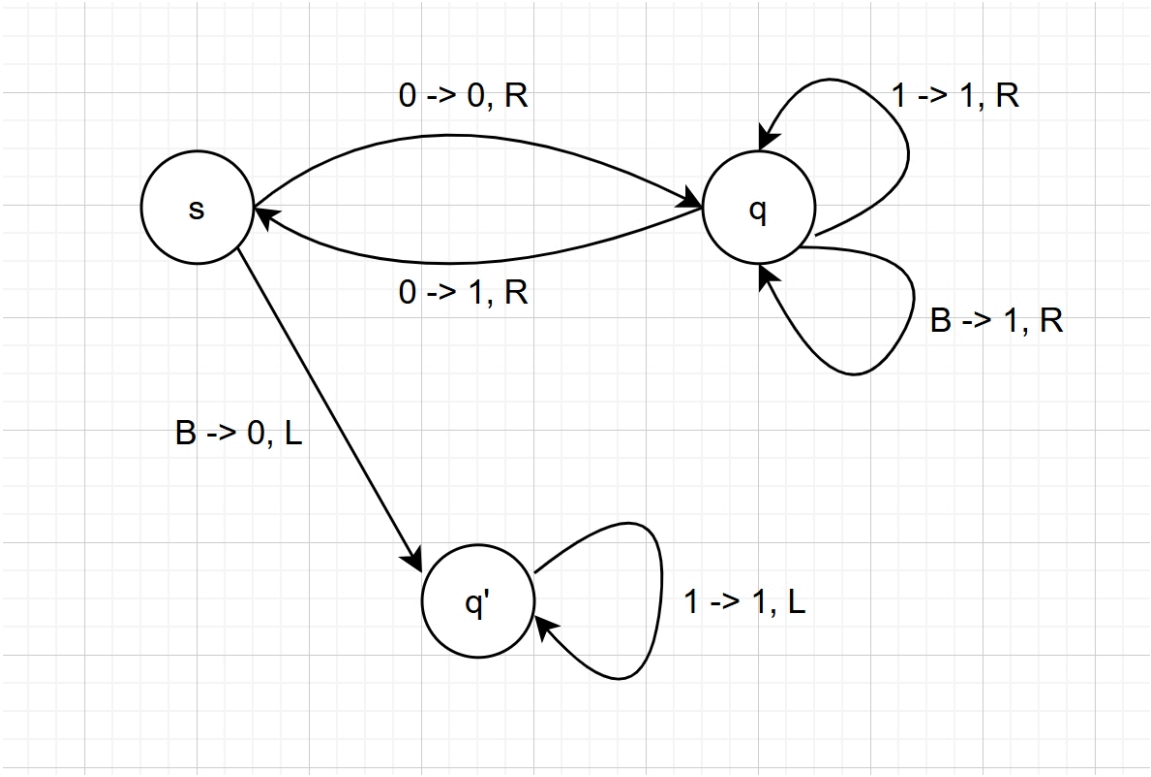
we first define the states set $Q = \{s, q, q'\}$ with

- s: the initial state
- q: moving to the left
- q': moving back to the first 0 just in case

we have the Turning Machine

$Q = \{s, q, q'\}, \sum = \{B, 0, 1\}, s = s, \delta =$

	B	0	1
s	$(q', 0, L)$	$(q, 0, R)$	
q	$(q, 1, R)$	$(s, 1, R)$	$(q, 1, R)$
q'			$(q', 1, L)$



then we have the computing process, we assume the starting position is the leftmost 0. When the initial tape is [1,2,3], the computation will be as follows, (I assume the current tape head is at the leftmost symbol of the third term in configuration, (a, 0,B1) would mean tape head is point at B)

the process can be described as follows, we iterate to the left most 0, changing every bit to 1, at the end, we insert an additional 1, then the state q' would move the tape head back to the initial 0

$$
\begin{aligned}
(s, B, 01^1B1^2B1^30B^2) &\to_M (q, B0, 1B1^2B1^30B^2)\\
&\to_M (q, B01, B1^2B1^30B^2)\\
&\to_M (q, 01^2, 1^2B1^30B^2)\\
&\to_M (q, 01^3, 1B1^30B^2)\\
&\to_M (q, 01^4, B1^30B^2)\\
&\to_M (q, 01^5, 1^30B^2)\\
&\to_M (q, 01^6, 1^20B^2)\\
&\to_M (q, 01^7, 10B^2)\\
&\to_M (q, 01^8, 0B^2)\\
&\to_M (s, 01^9, B^2)\\
&\to_M (q', 01^8, 10B)\\
&\to_M (q', 01^7, 1^20B)\\
&\to_M (q', 01^6, 1^30B)\\
&\to_M (q', 01^5, 1^40B)\\
&\to_M (q', 01^4, 1^50B)\\
&\to_M (q', 01^3, 1^60B)\\
&\to_M (q', 01^2, 1^70B)\\
&\to_M (q', 01, 1^80B)\\
&\to_M (q', 0, 1^90B)
\end{aligned}
$$

2 Exercise 2

When would you say that two RMs are computing the same?How would you define formally that two RMs compute in the same way? Is it decidable if two RMs are computing the same or computing in the same way, respectively? Argue why or why not?

I would say that two RM is computing the same(since is say, I dont think I need to formally define this)

they act the same, they both halts, and given any valid input, the output is always the same.I would consider that as long as they produce the same output on the same question, no matter what algorithms they use, no matter how twisted their program may be, they are computing the same

I would define formally that two RM is computing the same as

for any two RM: RM_1 and RM_2, they are considered computing the same if

first, they both halt, second, they use the same number of registers, third, given any configuration c, let the next configuration of RM_1 be fc_1 and that of RM_2 be fc_2, then $fc_1 = fc_2$.

they are both not decidable, because to decide whether they compute the same or in the same way, we have to decide whether they halt or not.

we do not know whether they halt or not because the halting question is known to be undecidable for arbitrary register machines, thus we cannot satisfy the requirements for both definitions

therefore, for both definitions, it is not decidable

3 Exercise 3

Describe graphically a Registe Machine (RM) M which sums up all the odd numbers up to a certain limit. More precisely, let

$$f(n) = \sum_{i=0}^{n} 2i + 1$$

then M should start with $R_0 = n$ and $R_1 = 0$ and terminate with $R_0 = 0$ and $R_1 = f(n)$

Write down the program code or list of instructions for this machine M using only a single HALT instruction (at the end of the code)

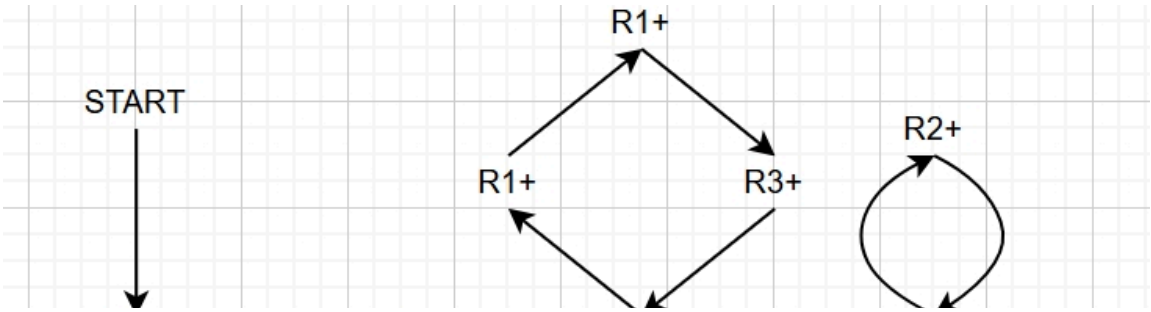
Describe the executions fo M with initially $R_0 = 0, 1, 2, 3$

Do all executions of M halt? Explain why or why not

the register machine starts with $R_0 = n$ and the rest 0, or initial configuration (0,n,0,0,0)

the Register machine should form a loop, if we are using curved lines, then the execution graph should be like a big main circle and some small circles just on the edge of the big circle and a main line tagent to the main circle pointing directly at the HALT

it should look like this



$$
\begin{aligned}
L_0 &: R_0^- \to L_1, L_9 \\
L_1 &: R_2^+ \to L_2 \\
L_2 &: R_2^- \to L_3, L_6 \\
L_3 &: R_1^+ \to L_4 \\
L_4 &: R_1^+ \to L_5 \\
L_5 &: R_3^+ \to L_2 \\
L_6 &: R_3^- \to L_7, L_8 \\
L_7 &: R_2^+ \to L_6 \\
L_8 &: R_1^+ \to L_0 \\
L_9 &: R_1^+ \to L_{10} \\
L_{10} &: HALT
\end{aligned}
$$

if $R_0 = 0$ then the loop would just add 1 to R_1 and HALT

if $R_0 = 1, 2, 3$, then the programs forms a loop, each time we add 1 to R_2, pass through a doubler to add $2 * R_2$ to R_1 and R_2 to R_3, we then use the R_3 to restore the value of R_2 back, we also add 1 to R_1 everytime in the loop for the +1 term. In addition, because $2 \times 0 + 1 = 1$ was not added in the loop, when R_0 is zero, we exit the loop directly, so we add this back at the end of the program

it also HALTs regardless of R_0

if $R_0 = 0$, the program would halt obviously, all it does is adding 1 to R_1

otherwise it also HALTS, no matter what value R_0 is, assume it to be n, the real variable R_0 in execution to be m, then the loop will executes for n times, R_2 in the begining of the loop would be n-m, the doubler would execute $4 \times (n - m)$ times, copying R_2 to R_3 take another $2 \times (n - m)$ times,

in each loop, there is an additional 4 executions beside the above, and 2 executions outside the loop

so the total executions would be, as m can be 1 to n-1 as R_0 is substracted by 1 immediately as we enter the loop so $2 + \sum_{m=0}^{n-1} 6 \times (n - m) + 4 = 3n(n + 1) + 4n + 2$ executions which is finite

so the program always HALTS