

Worth: 15%

1. [20 marks]

Show that the following functions are computable:

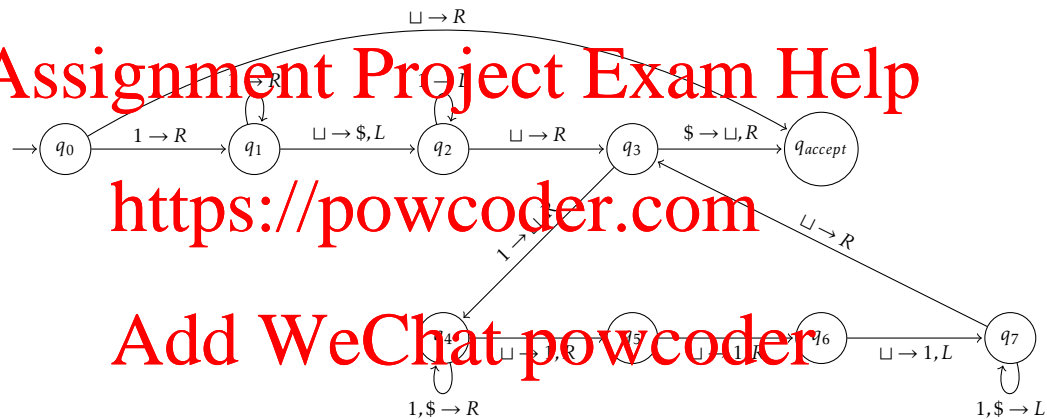
(a)  $f(x) = 3x$

(b)  $f(x, y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{otherwise.} \end{cases}$

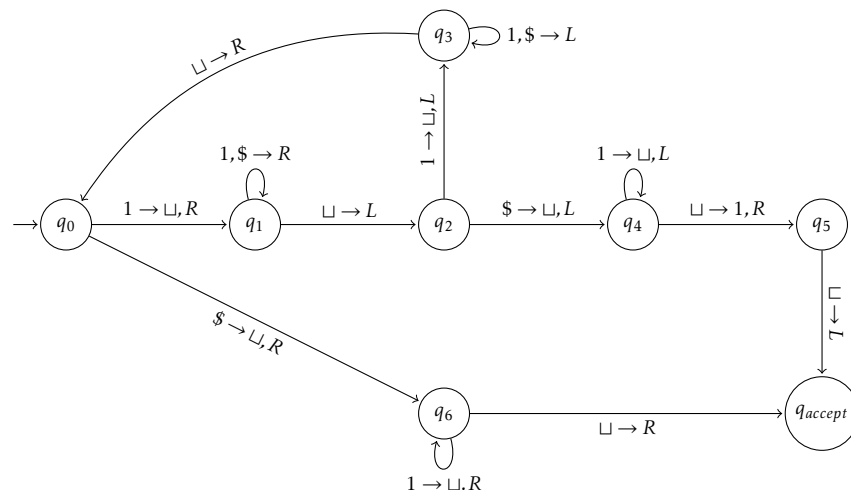
You can assume the input numbers are in unary, and for part (b), the input is given as  $x\$y$ .

**Solution.**

- (a) High level idea: put a delimiter \$ at the end of the input string; then, for each 1 read from the input, change that 1 to a blank, keep moving right past the delimiter to the rightmost end, put three 1s on the tape, move all the way left past the delimiter to the leftmost 1, and repeat until there are no more 1s left to the left of the delimiter. Change \$ to a blank, move right and halt.



- (b) High level idea: match a 1 from  $x$  with a 1 from  $y$  until no more 1s are left either on the  $x$  side or on the  $y$  side. If a 1 from  $x$  is not matched, erase everything and write 1; else erase everything (0 is equivalent to blank in unary).



2. [20 marks]

A *Turing machine with left reset* is a Turing machine with a singly infinite tape and the following transition function:

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, RESET\}.$$

If  $\delta(q, a) = (q', b, RESET)$ , then the Turing machine replaces  $a$  with  $b$  at its current tape position, moves the tape head to the leftmost position on the tape, and transitions from state  $q$  to state  $q'$ .

Show that the set of languages recognized by the class of Turing machines with left reset is the same as the set of languages recognized by the class of standard Turing machines.

**Solution.** It is clear that a Turing machine  $M'$  with left reset can be simulated by a Turing machine  $M$  with right and left moves only, so it suffices to show that a left move can be simulated by a combination of right moves and left reset to prove the claim. We will describe that  $M$  can be simulated by  $M'$  (except possibly using a larger tape alphabet) in the following way.

Suppose  $M$  wants to perform a transition  $\delta(q, a) = (q', b, L)$ . Then  $M'$  simulates  $M$  in state  $q$  when reading  $a$  on the tape by firstly writing a marked symbol  $\hat{b}$  at the current position, moving right to mark the end of the non-empty part of the tape with  $\#$ , and resetting. Next,  $M'$  uses new states to shift its current contents on its tape one cell to the right and remember the next state  $q'$  it should transition to. To do this,  $M'$  uses a state that remembers that the cell to the left of it contained  $x$  originally, it replaces  $b$  with  $\hat{x}$ . Furthermore, when  $\hat{b}$  is shifted to its immediate right, it is unmarked. Once the  $\#$  is replaced,  $M'$  resets and finds the marked symbol  $\hat{x}$  by scanning the tape from left to right. Once  $\hat{x}$  is found, the tape head of  $M'$  is now reading a symbol that is one cell left of its original position and  $M'$  remembers that it transitioned to state  $q'$  before the reset. Hence  $M'$  has performed a left move with right and reset only.

3. [20 marks]

Prove that a language  $A \subseteq \Sigma^*$  is semi-decidable if and only if there is a decidable binary relation  $R \subseteq \Sigma^* \times \Sigma^*$  such that for all  $x \in \Sigma^*$ ,

$$x \in A \text{ if and only if there is some } y \in \Sigma^* \text{ for which } (x, y) \in R.$$

Recall that a binary relation  $R$  is decidable if there is a Turing machine  $M$  that determines in finite time whether or not  $(x, y) \in R$  for a given pair  $(x, y) \in \Sigma^* \times \Sigma^*$ .

**Solution.** Assume a decidable relation  $R$  exists. Design a TM  $M$  for  $A$  as follows:

On input  $x$ :

- Generate strings  $y \in \Sigma^*$  in lexicographic order, and check if  $(x, y) \in R$ ; if yes, accept!

By the definition of  $R$ , it is clear that  $\mathcal{L}(M) = A$ , and hence,  $A$  is semi-decidable.

Conversely, if  $A \subseteq \Sigma^*$  is semi-decidable, then there is a Turing machine  $M$  recognizing  $A$ . We have to construct a decidable binary relation  $R$  such that  $x \in A$  if and only if there is some  $y$  with  $(x, y) \in R$ . Many choices of  $R$  are possible. One choice of relation  $R$  is the set  $(x, y)$  where  $y$  is the number of steps  $M$  takes to accept on input  $x$ . This number  $y$  can be encoded according to the alphabet  $\Sigma$  in unary (if  $a \in \Sigma$  is a symbol, the number  $n$  is represented by the string  $a^n \in \Sigma^*$ ). Furthermore, the relation  $R$  is decidable since a Turing machine can determine whether or not  $(x, y) \in R$  by running  $M$  for at most  $y$  steps, and then accepting if  $M$  has accepted  $x$  within that time (and rejecting otherwise).

4. [20 marks]

Show that a language  $L$  is decidable if and only if there is some enumerator  $E$  that prints the strings of  $L$  in lexicographic order.

**Solution.** Suppose  $L$  is finite. Then we can design an enumerator where we can "hardcode" all the strings of  $L$  and have it print them lexicographically. Conversely, we can show that  $L$  is decidable by designing a TM that has the strings of  $L$  hardwired in it and accepts any input iff it matches one of those strings. The decision procedure completely ignores the enumerator. So, we can assume without loss of generality that  $L$  is infinite.

Suppose  $L$  is decidable. Let  $M$  be a Turing machine that decides  $M$ . Then one can construct an enumerator for  $E$  by generating strings  $s_i$  in lexicographic order, running  $M$  on each string, and printing  $s_i$  iff  $M$  accepts  $s_i$ . Every string in the language  $L$  will be listed at some point since  $M$  halts on all inputs, and by construction  $E$  prints the strings in lexicographic order.

Conversely, suppose that there is an enumerator  $E$  that prints  $L$  in lexicographic order. Then one can decide in finite time whether a given string  $x$  is in the language  $L$  or not by running  $E$ , and accepting  $x$  if  $x$  is printed by  $E$ , or rejecting  $x$  once a string greater than  $x$  in the lexicographic order is printed. Thus,  $L$  is decidable.

**Assignment Project Exam Help**

**<https://powcoder.com>**

**Add WeChat powcoder**