Lab08-Computational Complexity

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2019.

$*$ If there is any problem, please contact TA Jiahao Fan or TA Mingran Peng.

$*$ Name:BowenZhang Student ID:517021910797 Email: 372799293@qq.com

1. Design a one-tape TM M that computes the function $f(x, y) = x - y$, where x and y are positive integers ($x > y$). The alphabet is $\{1, 0, \square, \triangleright, \triangleleft\}$, and the inputs are x 1's, $\square$ and y 1's. Below is the initial configuration for input $x = 7$ and $y = 3$. The result $z = f(x, y)$ should also be represented in the form of z 1's on the tape with the pattern of $\triangleright 111 \cdots 111 \triangleleft$.

Init:

	$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$
	$\uparrow$ q_S												

(a) Please describe your design and then write the specifications of M in the form like $\langle q_S, \triangleright\rangle \to \langle q_1, \triangleright, R\rangle$. Explain the transition functions in detail.

Solution. Firstly,we move the head to the symbol $\square$,and then we replace one 1 with $\square$ on both right side and left side of $\square$,which means that we reduce both x and y by 1.In the end,we will find that there is a $\triangleleft$ instead of 1 in the right side of rightmost $\square$,which means we have already replace all 1 in y and y's 1 in x with $\square$.At that time,we tape has the pattern of $\triangleright 111 \cdots 111 \square \cdots \square \triangleleft$.The only thing we need to do is move the $\triangleleft$ to the rightmost 1.

Start State:

$\langle q_S, \triangleright\rangle \to \langle q_M, \triangleright, R\rangle$

Move the head to the begin of y:

The head will move to right until it reads $\square$,and then we can change to the state q_R to start our algorithm.

$\langle q_M, 1\rangle \to \langle q_M, 1, R\rangle$

$\langle q_M, \square\rangle \to \langle q_R, \square, R\rangle$

Reduce y by 1:

The head will move to right until it doesn't reads $\square$.

$\langle q_R, \square\rangle \to \langle q_R, \square, R\rangle$

The head reads 1 means that y is still positive,and we reduce y by 1 through replacing 1 to $\square$.

$\langle q_R, 1\rangle \to \langle q_L, \square, L\rangle$

The head reads $\triangleleft$ means that y is 0 now and there are $x - y$ 1's on the tape with the pattern of $\triangleright 111 \cdots 111 \square \cdots \square \triangleleft$.And we need to change to state q_T to transform it to the pattern we desired.

$\langle q_R, \triangleleft\rangle \to \langle q_B, \square, L\rangle$

Reduce x by 1 correspondingly:

The head will move to left until it reads 1,and then replace 1 with $\square$ which means reduce x by 1,then turn to state q_R to reduce y.

$\langle q_L, \square\rangle \to \langle q_L, \square, L\rangle$

$\langle q_L, 1\rangle \to \langle q_R, \square, R\rangle$

Move back to the rightmost 1:

The head will move to left until it reads 1,then turn to the state q_T.

$\langle q_B, \square \rangle \to \langle q_B, \square, L \rangle$
$\langle q_B, 1 \rangle \to \langle q_T, 1, R \rangle$
Terminate:
$\langle q_T, \square \rangle \to \langle q_E, \triangleleft, S \rangle$

$\square$

(b) Please draw the state transition diagram using Microsoft Visio.

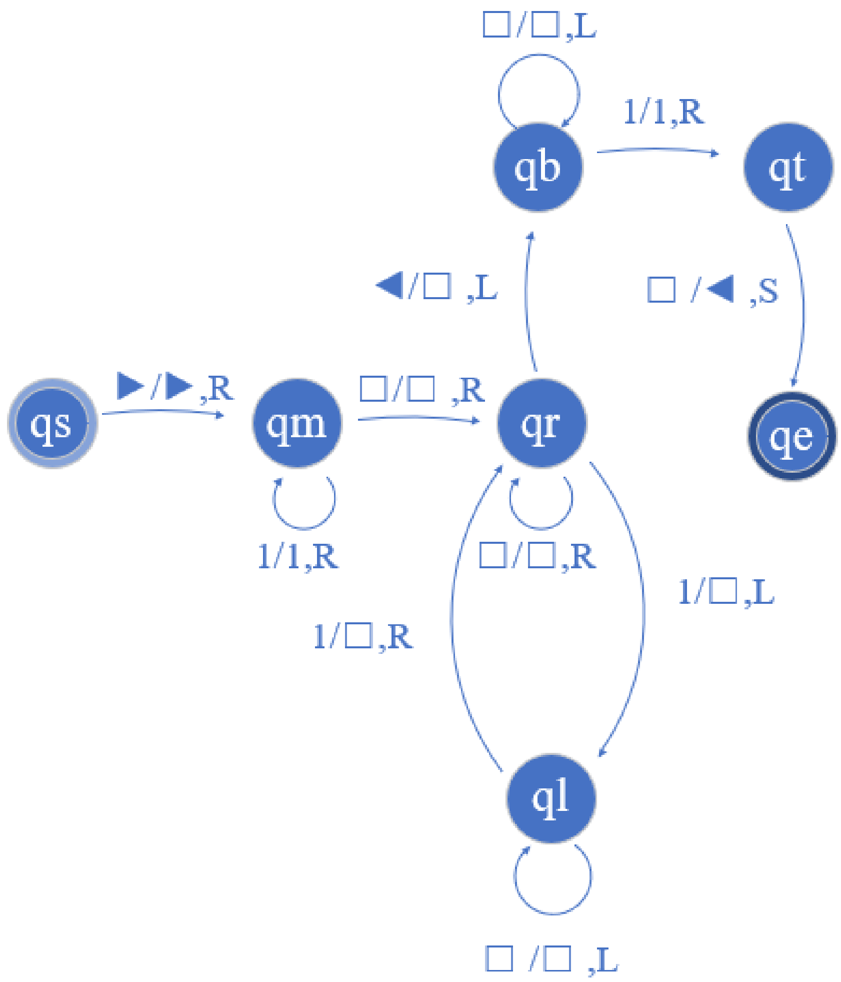

图 1: State transition diagram

(c) Show briefly and clearly the whole process from initial to final configurations for input $x = 7$ and $y = 3$.

	$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$	
	$\uparrow$ q_S													

	$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$	
		$\uparrow$ q_M												

$\cdots$ (Move to right)

	$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$	
									$\uparrow$ q_M					

$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$
									$\uparrow q_R$			

$\triangleright$	1	1	1	1	1	1	1	$\square$	$\square$	1	1	$\triangleleft$
								$\uparrow q_L$				

$\triangleright$	1	1	1	1	1	1	1	$\square$	$\square$	1	1	$\triangleleft$
							$\uparrow q_L$					

$\triangleright$	1	1	1	1	1	1	$\square$	$\square$	$\square$	1	1	$\triangleleft$
								$\uparrow q_R$				

$\cdots$ (Move to right)

$\triangleright$	1	1	1	1	1	1	$\square$	$\square$	$\square$	1	1	$\triangleleft$
										$\uparrow q_R$		

$\triangleright$	1	1	1	1	1	1	$\square$	$\square$	$\square$	$\square$	1	$\triangleleft$
									$\uparrow q_L$			

$\cdots$ (Move to left)

$\triangleright$	1	1	1	1	1	1	$\square$	$\square$	$\square$	$\square$	1	$\triangleleft$
						$\uparrow q_L$						

$\triangleright$	1	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	1	$\triangleleft$
							$\uparrow q_R$					

$\cdots$ (Move to right)

$\triangleright$	1	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	1	$\triangleleft$
											$\uparrow q_R$	

$\triangleright$	1	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\triangleleft$
										$\uparrow q_L$		

$\cdots$ (Move to left)

$\triangleright$	1	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\triangleleft$
					$\uparrow q_L$							

$\triangleright$	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\triangleleft$
						$\uparrow q_R$						

$\cdots$ (Move to right)

$\triangleright$	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\triangleleft$
												$\uparrow$ q_R

$\triangleright$	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$
											$\uparrow$ q_B	

$\cdots$ (Move to left)

$\triangleright$	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$
				$\uparrow$ q_B								

$\triangleright$	1	1	1	1	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$
					$\uparrow$ q_T							

$\triangleright$	1	1	1	1	$\triangleleft$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$
					$\uparrow$ q_E							

2. What is the “certificate” and “certifier” for the following problems?

   (a) *PARTITION*: Given a finite set A and a size $s(a) \in \mathbb{Z}$ for each $a \in A$, is there a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) = \sum_{a \in A-A'} s(a)$?

   Certificate: A subset A'

   Certifier: Check if $\sum_{a\in A'} s(a) = \sum_{a \in A-A'} s(a)$.

   (b) *CLIQUE*: Given a graph $G = (V, E)$ and a positive integer $K \leq |V|$, is there a subset $V' \subseteq V$ with $|V'| \geq K$ such that every two vertices in V' are joined by an edge in E ?

   Certificate: A subset V' with $|V'| \geq K$.

   Certifier: Check that if every two vertices in V' are joined by an edge in E.

   (c) *ZERO-ONE INTEGER PROGRAMMING*: Given an integer $m \times n$ matrix A and an integer m-vector b, is there an integer n-vector x with elements in the set $\{0, 1\}$ such that $Ax \leq b$?

   Certificate: An integer n-vector x with elements in the set $\{0, 1\}$.

   Certifier: Check that if $Ax \leq b$.

3. *SUBSET SUM*: Given a finite set A, a size $s(a) \in \mathbb{Z}$ for each $a \in A$ and an integer B, is there a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) = B$?

   KNAPSACK: Given a finite set A, a size $s(a) \in \mathbb{Z}$ and a value $v(a) \in \mathbb{Z}$ for each $a \in A$ and integers B and K, is there a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) \leq B$ and $\sum_{a\in A'} v(a) \geq K$?

   (a) Prove *PARTITION* $\leq_p$ *SUBSET SUM*.

   Proof. Instance of *PARTITION*:Given set A, a size $s(a) \in \mathbb{Z}$ for each $a \in A$.

   Instance of *SUBSET SUM*:Same set A,a size $s'(a) = 2s(a) \in \mathbb{Z}$ for each $a \in A$,$B = \sum_{a\in A} s(a)$.

   To prove:

i. If there is a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) = \sum_{a\in A-A'} s(a)$,then $\sum_{a\in A'} s'(a) = \sum_{a\in A'} 2s(a) = \sum_{a\in A} s(a) = B$

ii. If there is a subset $A' \subseteq A$ such that $B = \sum_{a\in A} s(a) = \sum_{a\in A'} s'(a) = \sum_{a\in A'} 2s(a)$,then $\sum_{a\in A-A'} s(a) = \sum_{a\in A} s(a) - \sum_{a\in A'} s(a) = B - \frac{1}{2}B = \frac{1}{2}B = \sum_{a\in A'} s(a)$.

□

(b) Prove *SUBSET SUM* $\leq_p$ *KNAPSACK*.

Proof. Instance of *SUBSET SUM*:A finite set A, a size $s(a) \in \mathbb{Z}$ for each $a \in A$ and an integer B.

Instance of *KNAPSACK*:Same set A, same size $s(a) \in \mathbb{Z}$ for each $a \in A$, a value $v(a) = s(a) \in \mathbb{Z}$ for each $a \in A$,same integer B,an integer K=B.

To prove:

i. If there is a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) = B$,then we have $\sum_{a\in A'} s(a) = B$ and $\sum_{a\in A'} v(a) = K$.

ii. If there is a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) \leq B$ and $\sum_{a\in A'} v(a) \geq K$,then we have $\sum_{a\in A'} s(a) = B$.

□

4. *3-SAT*: Given a set U of variables, a collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$, is there a satisfying truth assignment for C?

Prove *3-SAT* $\leq_p$ *CLIQUE*.

Proof. Instance of *3-SAT*:A set U of variables, a collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$ and $|C| = K$.

Instance of *CLIQUE*:An integer K and construct a graph $G = (V, E)$ as following:

(a) G contain 3 vertices for each clause,one for each literal.

(b) There is no edge between the literal in the same clause.

(c) Connect literal to the literal in other clause which has no contradiction with itself.

For example,if there are two clause which is $\overline{x_1} \vee x_2 \vee x_3$ and $x_1 \vee \overline{x_2} \vee x_3$.The $\overline{x_1}$ in clause 1 will connect to $\overline{x_2}$ and x_3 in clause 2 for they have no contradiction,and it will not connect to $\overline{x_1}$ in clause 2 for they have contradiction.

To prove:

(a) If there is a satisfying truth assignment for C,it means that we can find at least one literal in each clause which is satisfied requirement,and we random pick one satisfied literal in each clause,they don't contradict with each other.Thus,we can find at least K vertex in G,every two vertices of them are joined by an edge.

(b) If there is a subset $V' \subseteq V$ with $|V'| \geq K$ such that every two vertices in V' are joined by an edge in E,then the vertex in $|V'|$ come from different clause for the vertex in the same clause will not connect to each other.Thus,we can find a literal in each clause,they don't contradict with each other,which means it is a satisfying truth assignment for C.

□

5. Algorithm class is a democratic class. Denote class as a finite set S containing every students. Now students decided to raise a student union $S' \subseteq S$ with $|S'| \leq K$.

   As for the members of the union, there are many different opinions. An opinion is a set $S_o \subseteq S$. Note that number of opinions has nothing to do with number of students.

   The question is whether there exists such student union $S' \subseteq S$ with $|S'| \leq K$, that S' contains at least one element from each opinion. We call this problem *ELECTION* problem, prove that it is NP-complete.

   Proof. Firstly,we need to prove that *ELECTION* is a NP problem:

   Certificate: A student union $S' \subseteq S$ with $|S'| \leq K$.

   Certifier: Check if S' contains at least one element from each opinion.And we can easily find that it can be done in polynominal time.

   Secondly,we try to prove that *SET COVER* $\leq_p$ *ELECTION*:

   Instance of *SET COVER*:A set U of elements,a collection $S_1, S_2, \ldots, S_m$ of subsets of U,and an integer K.

   Instance of *ELECTION*:Same set U of opinions,same collection $S_1, S_2, \ldots, S_m$ of subsets of U,subset S_i represent the opinions that i^{th}student hold.From that,we can easily construct S_o for each opinion.A finite set S containing m students.Same integer K.

   (a) If there exist a collection of $\leq k$ of these subsets whose union is equal to U,we denote these subsets as $S_{i_1}, S_{i_2}, \ldots, S_{i_n}$,$n \leq k$. Then we have a student union $S' \subseteq S$,S' containing student i_1,student i_2,...,student i_m,$m \leq K$,those student cover all opinions,thus S' contains at least one element from each opinion.

   (b) If there exists such student union $S' \subseteq S$ with $|S'| \leq K$, that S' contains at least one element from each opinion,which means the students in S' cover all the opinions.Suppose S' containing student i_1,student i_2,...,student i_m,$m \leq K$,then the union of $S_{i_1}, S_{i_2}, \ldots, S_{i_n}$,$n \leq k$ is equal to U.

       As *SET COVER* is NP-complete,*ELECTION* is NP and *SET COVER* $\leq_p$ *ELECTION*,we can induce that *ELECTION* is NP-complete.

   □

Remark: You need to include your .pdf and .tex files in your uploaded .zip file.