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:田雪飞 Student ID:515030910347 Email: 13487426939@qq.com

1. **Solution.** The following is the process of solution.
a. In this case, the set of state is $Q = \{q_S, q_1, q_2, q_3, q_4, q_H\}$.
we use q_1 to find $\Box$, then convert q_1 to q_2. If q_2 find 1 from left to right, then convert q_2 to q_3 and convert 1 to $\Box$, turn direction around. If q_3 find 1 from right to left, then convert q_3 to q_2 and convert 1 to $\Box$, turn direction around, ; Repeat until q_2 find $\triangleleft$ then turn direction around, and convert q_2 to q_4, Then we use q_4 to find 1 and turn direction aroundp and convert q_4 to q_H.
Begin: use q_1 to find $\Box$, and convert q_1 to q_2, q_2 point to 1 on the left of $\Box$.
$\langle q_S, \triangleright\rangle \rightarrow \langle q_1, \triangleright, R\rangle$.
$\langle q_1, 1\rangle \rightarrow \langle q_1, 1, R\rangle$.
$\langle q_1, \Box\rangle \rightarrow \langle q_2, \Box, R\rangle$.
Loop: use q_2, q_3 to convert 1 to $\Box$ one by one alternately.
$\langle q_2, \Box\rangle \rightarrow \langle q_2, \Box, R\rangle$.
$\langle q_2, 1\rangle \rightarrow \langle q_3, \Box, L\rangle$.
$\langle q_3, \Box\rangle \rightarrow \langle q_3, \Box, L\rangle$.
$\langle q_3, 1\rangle \rightarrow \langle q_2, \Box, R\rangle$.
End: use q_4 to find 1 on the left of $\Box$ and p_4 point to next cell.
$\langle q_2, \triangleleft, \rangle \rightarrow \langle q_4, \triangleleft, L\rangle$.
$\langle q_4, \Box\rangle \rightarrow \langle q_4, \triangleleft, L\rangle$.
$\langle q_4, 1, \rangle \rightarrow \langle q_H, 1, R\rangle$.

b.The state transition diagram in txf_homework_08.vsdx like following picture.

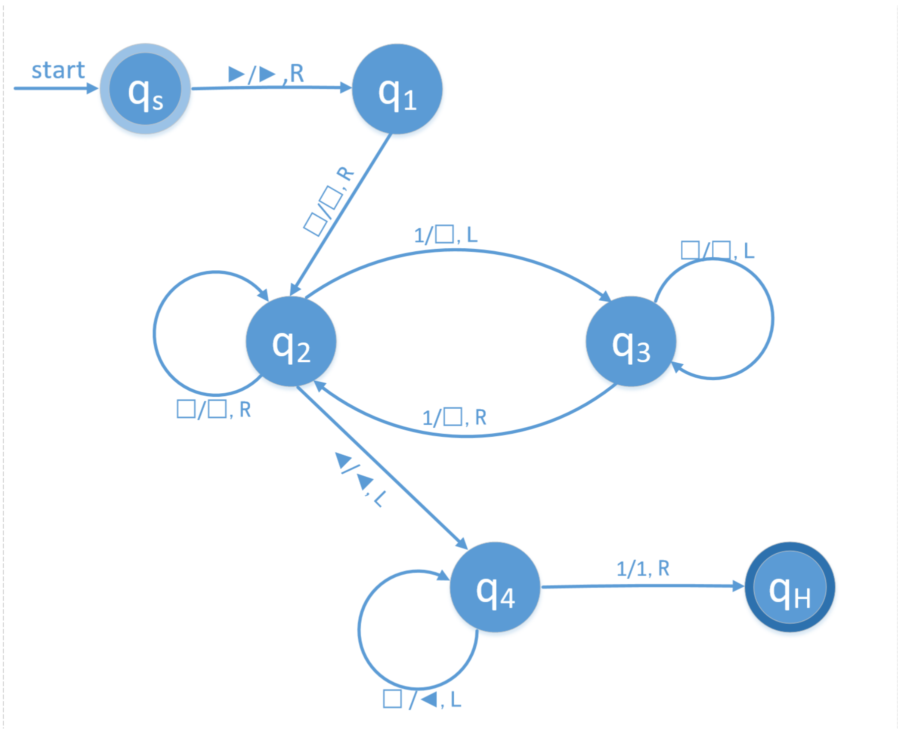


Figure 1: The state transition diagram

c.Whole process from initial to final configurations.

$\triangleright$	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_1											

$\vdots$

$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$
								$\uparrow$ q_1				

$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$
									$\uparrow$ q_2			

$\triangleright$	1	1	1	1	1	1	1	$\square$	$\square$	1	1	$\triangleleft$
								$\uparrow$ q_3				

$\triangleright$	1	1	1	1	1	1	1	$\square$	$\square$	1	1	$\triangleleft$
							$\uparrow$ q_3					

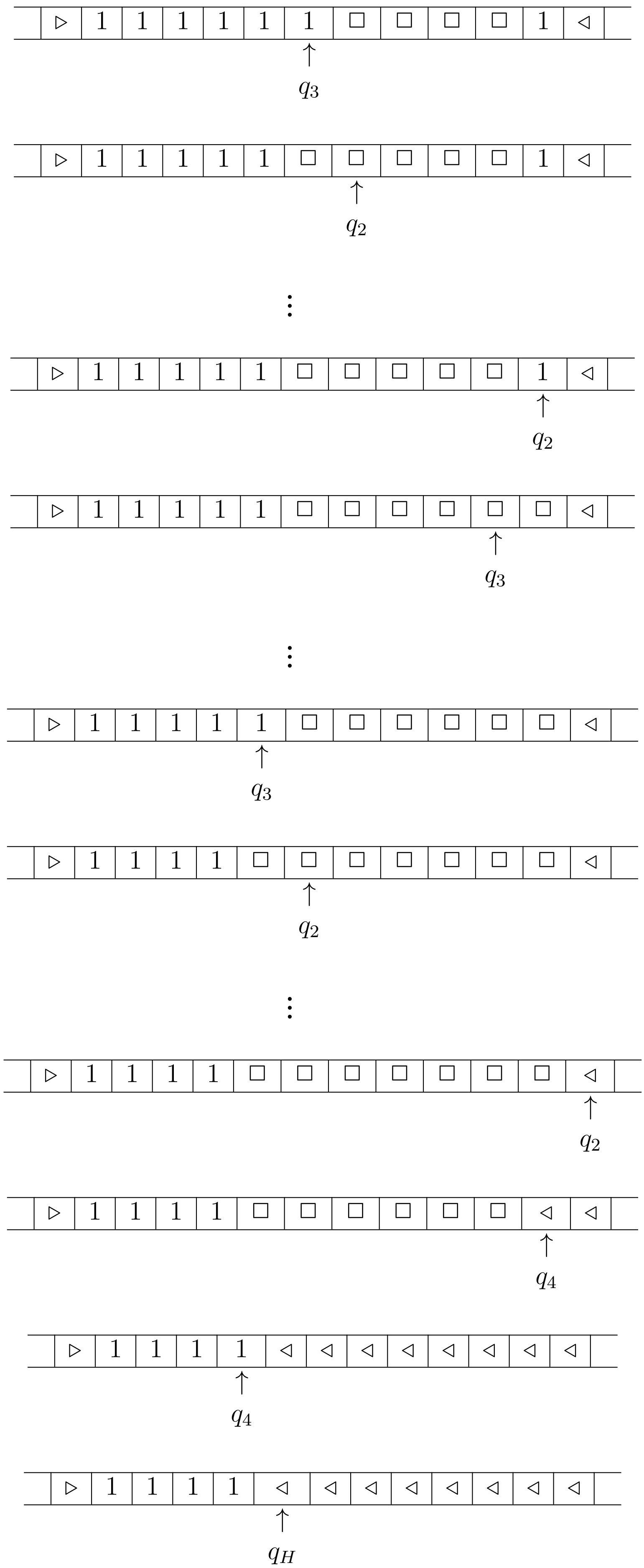
$\triangleright$	1	1	1	1	1	1	$\square$	$\square$	$\square$	1	1	$\triangleleft$
								$\uparrow$ q_2				

$\triangleright$	1	1	1	1	1	1	$\square$	$\square$	$\square$	1	1	$\triangleleft$
									$\uparrow$ q_2			

$\triangleright$	1	1	1	1	1	1	$\square$	$\square$	$\square$	1	1	$\triangleleft$
										$\uparrow$ q_2		

$\triangleright$	1	1	1	1	1	1	$\square$	$\square$	$\square$	$\square$	1	$\triangleleft$
									$\uparrow$ q_3			

$\vdots$



$\square$

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 $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$?

   (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 ?

   (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 $Ax \leq b$?

   Solution. The following is process of solution.
   a. Certificate: Note that such a certificate exist only if $\sum_{a \in A'} s(a)$ is equal to $\frac{1}{2} \sum_{a \in A} s(a)$; example: $A = 1, 2, 1, 1, 2, 3, 5, 6, 7, 8$; $A' = 1, 2, 1, 2, 3$.
   Certifier: convert the size of each $a \in A$ to a sequence $s(a_1), s(a_2), \ldots, s(a_n)$ and sort the sequence;

   Algorithm 1: certifier

   Input: A sorted sequence $s(a_1), s(a_2), \ldots s(a_n)$
   Output: Yes or No
   $sum \leftarrow 0$;
   if $\sum_1^n = odd$ **then**
       return *false*;
   else
       for *i to n* **do**
           sum=0;
           for *j to n* **do**
               $sum+ = s(a_j)$
               if $sum = \frac{1}{2} \sum_{a \in A} s(a)$ **then**
                   return *true*
   return *false*

   ---

   b. Certificate: Check every k vertices and judge whether every two vertices are connected with each other;
   Certifier: There exist k vertices in which every two vertices are connected.
   c. Certificate:A m $\times$ n matrix A and an integer m-vector b and an integer n-vector x.
   Certifier: $Ax \leq b$. ☐

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) **Proof.** Prove *PARTITION* $\leq_p$ *SUBSET SUM.*
   In the PARTITION problem: $\sum_{a \in A'} s(a) + \sum_{a \in A-A'} s(a) = \sum_{a \in A} s(a)$, and we assume $\sum_{a \in A} s(a) = N$;
   So we can convert $\sum_{a \in A'} s(a) = \sum_{a \in A-A'}$ to $\sum_{a \in A} s(a) = \frac{N}{2}$, then let $B = \frac{N}{2}$;
   So, we can reduces PARTITION to SUBSET SUM. ☐

(b) **Proof.** Prove *SUBSET SUM* $\leq_p$ *KNAPSACK*.
In SUBSET SUM problem: we construct the value of $a \in A$ is v(a) and make v(a)=s(a), and make $B = K$.
so,we can reduce the question (is there a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) = B$) to the question (is there a subset $A' \subseteq A$ such that $\sum_{a\in A'} s(a) \leq B$and $\sum_{a\in A'} s(a) \geq K.(K = B)$).
So, we can reduces SUBSET SUM to KNAPSACK.

□

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. We prove this question by two steps.
   First step: prove 3-SAT reduces to independent set.
   Proof: Given an instance Φ of 3-SAT, we construct an instance (G,K) of independent-set that has an independent set of size k iff Φ is satisfiable.
   construct:
   • G contains 3 vertices for each clause,one for each literal.
   • Connect 3 literals in a clause in a triangle.
   • Connect literal to each of its negations.

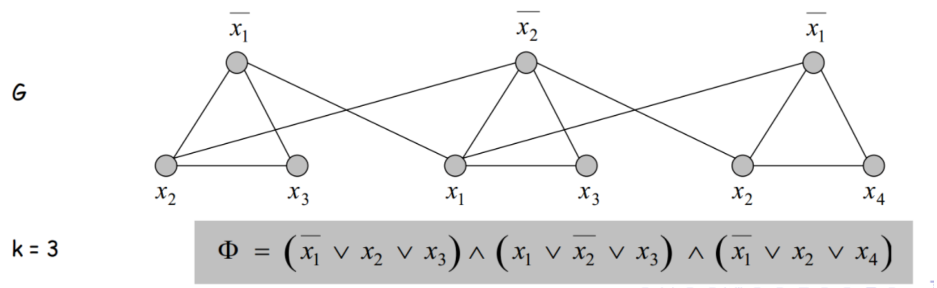


Figure 2: 3-SAT and Independent set graph.

   proof: $\Rightarrow$ Let S be independent set of size k.
   • S must contain exactly one vertex in each triangle.
   • Set these literals to true and any other variables in a consistent way.
   •Truth assignment is consistent and all clauses are satisfied
   proof: $\Leftarrow$ Given satisfying assignment, select one true literal from each triangle. This is an independent set of size k.
   Second step: prove Independent set to clique.
   construction: we construct the Complementary graph of G is $\overline{G}$.we should prove that if G exists an independent set of size k iff $\overline{G}$ exists an clique of size k.
   proof: $\Rightarrow$ If G exists an independent set of size k, then there exists k vertices have no edge in them, obviously, These k vertices must be joined by an edge between every two vertices in $\overline{G}$, This is a clique of size k in $\overline{G}$.
   proof: $\Leftarrow$ If $\overline{G}$ exists clique with size k.then there exists k vertices must be joined by a edge between every two vertices.Obvoiusly.G must have k vertices which have no edge between any two vertices. so, G have an independent set of size k.

   Finally: according to the step 1: *3-SAT* $\leq_p$ *Independent Set*, step 2: *Independent Set* $\leq_p$ *CLIQUE*. So *3-SAT* $\leq_p$ *CLIQUE*. □

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.

   Solution. Assume the number of students are N; Number them as $x_1, x_2, x_3 \ldots x_n$;and construct a set $\{x_1, x_2, x_3 \ldots x_n\}$, Assume there have M opinions, for example: $n = 5$; and opinions: 1:$\{x_1, x_2\}$,2:$\{x_2, x_4\}$,3:$\{x_3, x_4, x_5\}$; so $S' = \{x_2, x_4\}$ is an example which is satisfied condition. so we can reduce this problem to a SET-COVER problem.
   proof: ELECTION $\leq_p$ SET-COVER.
   Convert: we can number every opinions,for example:opinion 1:$\{x_1, x_2\}$,opinon2:$\{x_2, x_4\}$, opinion3:$\{x_3, x_4, x_5\}$; and then we convert it to student set:$x_1 : \{1\}$,$x_2 : \{1, 2\}$,$x_3 : \{3\}$, $x_4 : \{3\}$,$x_5 : \{3\}$,the aim is to find sets to cover all opinions. then we just need to find a SET-COVER to solve this problem.So we can convert this question to whether there exist a SET-COVER?
   So, ELECTION problem is a NP problem. $\square$

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