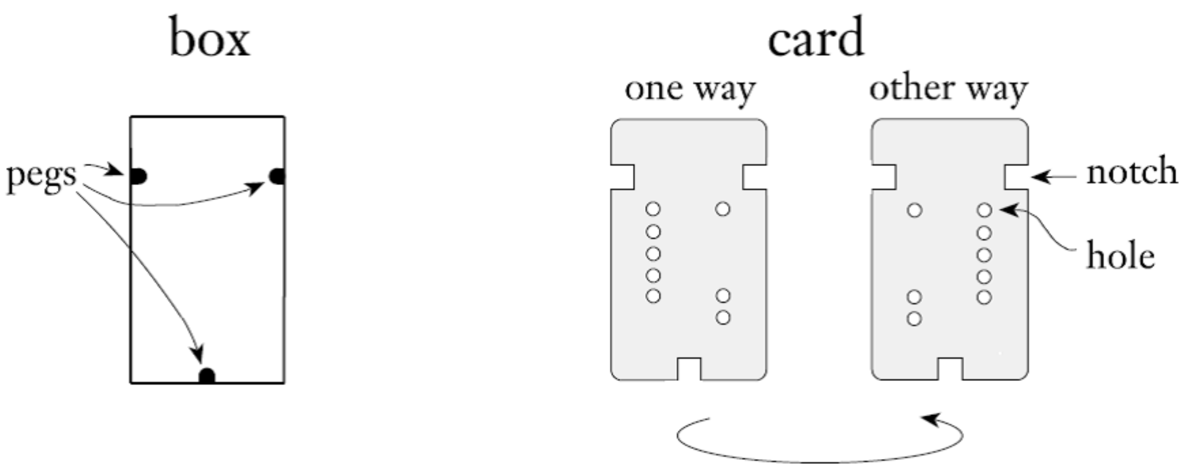
Problem

You are given a box and a collection of cards as indicated in the following figure. Because of the pegs in the box and the notches in the cards, each card will fit in the box in either of two ways. Each card contains two columns of holes, some of which may not be punched out. The puzzle is solved by placing all the cards in the box so as to completely cover the bottom of the box (i.e., every hole position is blocked by at least one card that has no hole there).

Let $PUZZLE = \{\langle c_1, \ldots, c_k \rangle |$ each c_i represents a card and this collection of cards has a solution}. Show that *PUZZLE* is NP-complete.



Step-by-step solution

Step 1 of 4

A collection of cards with notches is to be fitted into a box with pegs. A card may fit into the box in one of two ways. Each card has two vertical columns of holes. If the cards are all placed so as to completely cover the bottom of the box the puzzle is solved.

Therefore, it can be prove that $PUZZLE$ is NP-complete where $PUZZLE$ is defined as:

$$PUZZLE = \{\langle c_1, \ldots, c_k \rangle \mid \text{each } c_i \text{ represents a card and this collection has a solution}\}$$

• It can be observed that the order in which the cards are placed does not affect a solution to $PUZZLE$.

• First show the problem is in NP. Next NP-complete problems are proven to be polynomial time reducible to $PUZZLE$. This will prove that $PUZZLE$ is NP-complete.

Comment

Step 2 of 4

A verifier V for $PUZZLE$ will take the directions $d = d_1, \ldots, d_k$ in which the cards $c = c_1, \ldots, c_k$ are placed as the certificate.

The verifier V is:

$V =$ "On input $\langle c, d \rangle$:

1. Place the cards c in directions d in the box.
2. Test if the bottom of the box is completely covered.
3. If yes, *accept*; else, *reject*."

• It can be seen that all the steps of the verifier take polynomial time because the placing of card in step 1 will take a polynomial time.

• Also, it will take a polynomial time to test whether the box is completely covered or not, because user have to traverse all the bottom of the box.

Hence this problem is NP.

Comment

Step 3 of 4

To show that NP-complete problems are polynomial time reducible to $PUZZLE$.

For each and every instance of the 3SAT problem for a Boolean formula ϕ, it is shown how to construct an instance of the $PUZZLE$ problem.

• An instance ϕ of the 3SAT problem for a Boolean formula ϕ with l clauses and 3 variables in a clause will be as follows.

• It can have $l \cdot 3$ variables, which are $x_1, \ldots, x_{3l}$.

$\phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_l \vee b_l \vee c_l)$, where each a_j, b_j, c_j represents x_i or $\overline{x_i}$.

• For reducing an instance of the 3SAT problem into an instance of $PUZZLE$, the variable gadget and the clause gadget have to be created for k cards with columns of l holes on each card.

Comment

Step 4 of 4

Looking at the condition for $PUZZLE$ to be satisfied, it is that all the punched out holes need to be covered. Thus, the clause gadget will be the condition for **a hole to be filled by a stack of cards.**

• Next, observe that a clause gadget is true when any of the three variables it contains are true.

• On this basis the variable gadget is made to be the placement of three cards chosen from the stack of cards such that the bottom of the box is not visible from a particular hole.

• This **reduction** f **works in polynomial time for all instances of an NP-complete** problem, 3SAT. The proven has also been proven to lie in NP.

Therefore the problem $PUZZLE$ **is NP-complete**

Comment