

[03–60–454] Assignment 4

Stephen Nusko
103693282

April 13, 2015

1 Question 1

Given a deck of $n + 1$ cards each of which is imprinted with two columns of circles along the two sides. Each column contains m circles and some of the circles are punched out to make holes so that one can see through. One of the cards, called the TRUE card, has all the circles in one column punched out and all the circles in the other column left intact.

Determine if there is a way of stacking up the cards with the TRUE card on top so that all the holes are covered (i.e. no one can see through). Note that the two faces of each card are indistinguishable (i.e. you may flip the card over).

1.1 Part 1

Nondeterministic-Algorithm card_stacking

Input: A deck of n cards, plus one TRUE card.

Output:

$$\begin{cases} \text{yes} & \text{if all holes are covered by at least one card.} \\ \perp & \text{otherwise} \end{cases}$$

We adopt the following notation for this algorithm.

$c_i.hole[j]$ means the i^{th} card and its j^{th} hole.

circle_covered returns true if the hole passed in is not punched out, and false otherwise.

1. (The guessing phase)

Write down a sequence of cards c_1, c_2, \dots, c_n facing anyway, and then place the TRUE card on top.

2. (The verification phase)

let $open = \{ \text{idx}(\text{holes}) \mid \text{TRUE card doesn't cover it} \}$.

That is $open$ is the set of indicies of the holes which are not covered by the TRUE card.

for $i \leftarrow 1$ to $|open|$ do

 for $k \leftarrow 1$ to n do

 if (circle_covered($c_k.hole[open[i]]$)) then

 break, and continue next outer loop

 else if ($k == n$) then

 STOP

 endif

 end for

end for

Output('yes')

Step 1: $O(n)$ time, because we have to write down each card's position.

Step 2: $O(mn)$ time, because the TRUE card covers m holes leaving m holes uncovered, and we then loop through the uncovered holes, looking in the worst case at all n cards.

Total time: $O(mn)$

Hence, $\Pi \in \mathbf{NP}$. \square

1.2 Part b

Next we prove that $3\text{-SAT} \propto \Pi$.

Let $C = c_1 \wedge c_2 \wedge \dots \wedge c_m$ be a 3-CNF expression in which the c_j 's are the clauses, and $U = \{u_1, u_2, \dots, u_n\}$ be the set of variables in C .

For each variable u_i we create a $card_i$ with two columns of m circles (remember m is the number of clauses). For each row r_k if u_i or \bar{u}_i appears in k^{th} clause we punch out the left hole if it was u_i and the right hole if it was \bar{u}_i .

Otherwise if neither appears we punch both holes out.

Finally we create the TRUE card by creating an additional card with one column completely punched out.

Example

Given $(u_1 \vee u_2 \vee u_3) \wedge (u_4 \vee \bar{u}_1 \vee \bar{u}_5)$

Would result in

u_1	u_2	u_3	u_4	u_5	TRUE																								
<table><tr><td>●</td><td>○</td></tr><tr><td>○</td><td>●</td></tr></table>	●	○	○	●	<table><tr><td>●</td><td>○</td></tr><tr><td>●</td><td>●</td></tr></table>	●	○	●	●	<table><tr><td>●</td><td>○</td></tr><tr><td>●</td><td>●</td></tr></table>	●	○	●	●	<table><tr><td>●</td><td>●</td></tr><tr><td>●</td><td>○</td></tr></table>	●	●	●	○	<table><tr><td>●</td><td>●</td></tr><tr><td>○</td><td>●</td></tr></table>	●	●	○	●	<table><tr><td>●</td><td>○</td></tr><tr><td>●</td><td>○</td></tr></table>	●	○	●	○
●	○																												
○	●																												
●	○																												
●	●																												
●	○																												
●	●																												
●	●																												
●	○																												
●	●																												
○	●																												
●	○																												
●	○																												

With the previous process we have taken a instance C of a 3-CNF expression, and formed a instances of Π call it P .

Next we show that,

C is satisfiable iff K is a yes instance.

Prove that if C is satisfiable then K is a yes instance

Let $t : U \rightarrow \{true, false\}$ be a truth assignment satisfying C .

then $(\forall j) 1 \leq j \leq m, c_j \equiv true$ under t .

We construct an instance of Π as described above, and we consider each clause one by one.

Since the first clause $c_1 \equiv true$, we know that there is a variable u_i which caused this clause to be true, select the card that is associated with that variable. Without loss of generality in regard to its form as either u_i or \bar{u}_i flip the card so that the unpunched circle covers the left column.

For all future clauses we have two cases in regards to the variable that made the clause $true$.

We have seen that variable before and already placed its card, or it is a new unseen card.

Case 1: We have seen it before.

In this case we have seen this variable before, and it caused a previous clause to be evaluated to true. This variable is either of the same form or the negative of it. Without loss of generality assume that the first occurrence was of the form u_i .

Sub-Case 1: The current form is \bar{u}_i

In this case $u_i \equiv true$, so $\bar{u}_i \equiv false$, however we picked \bar{u}_i because it made this clause true. This is a contradiction, therefore this case can not occur.

Sub-Case 2: The current form is u_i .

Since the form is the same as the previous time we have the $u_i \equiv \text{true}$.

And by the process described since the variable is of the same form, the same side of the column was punched out for their respective holes.

Since we selected the card to cover the left in the past, and the hole is on the same side as the past, we have also covered this row's left hole.

Therefore in this case we cover this clause's respective left column.

Case 2: We haven't seen it before.

In this case we have selected the variable u_i (in either form) because it occurs in C and was true in the clause c_i . Again like the first case, we know that only one hole is punched out on row i . We therefore flip the card in such a way that the unpunched circle is on the left column.

Since we haven't seen this variable before, we haven't used this card, and placing it will not change any previous card placements.

Therefore in this case we cover this clauses's respective left column.

Therefore in both cases we were able to place the cards so each clauses's respective left column was covered. We repeat this for each clause, until all clause's left column are covered.

Any left over cards are then placed on top, and finally we place the TRUE card on top such that the unpunched circles are on the right.

Since we have covered each hole on the left by our placements of the n cards for each variable, and we have covered each hole on the right by the placement of the TRUE card, all holes are covered. Therefore K is a yes instance of Π .

Therefore if C is satisfiable then K is a yes instance of Π . \square

Prove that if K is yes instance then C is satisfiable

Now we must prove the converse, that a yes instance of Π implies C is satisfiable.

We use a proof by contraposition, that is if C is not satisfiable then K is a no instance of Π .

Assume that C is not satisfiable.

Then \forall truth assignments $t : U \rightarrow \{\text{true}, \text{false}\}$ t does not satisfy C .

let t be any truth assignment, then $(\exists j) 1 \leq j \leq m, c_j \equiv \text{false}$ under t .

Following the same procedure as done in the true case, for all true clauses we can cover the left column, so we now consider only the false clauses. One at a time.

If any of the false clauses contain a variable we haven't seen before. then in this case the variable that is unseen before is false because it is part of a false clause of ors. However this variable has not been used in any clauses which are true because we would have seen it when dealing with the true clauses.

Therefore it only occurs in false clauses. By negating this variable we make any clause in which it appears true, and because it doesn't appear in true clauses and leaves them unchanged.

However there must be other clauses which are false, because C was not satisfiable, and if only these false clauses existed, then negating the variable that is associated with this particular card would be a truth assignment which satisfied C , a contradiction.

Therefore there are more false clauses.

This leaves only the case of when all variables have been seen before.

In this case all variables have been seen before, and are therefore oriented so that the true clauses cover the left column.

This means that all the cards have holes aligned with the left column, and since no other variables occur in this clause all other cards will also have holes in the left column.

This proves that the only way to generate a hole in the left column of the stack of cards is to have all three variables be false, because if one was (true) or (false and not seen before), by the process described above we could cover the hole. Since we place the TRUE card so that it covers all the right column, but has holes on the left this will leave us with a hole in the instance K of Π .

To correct this in K we must flip one of the cards so that it covers the hole on the left.

We consider two cases either we can flip the card and no new holes on the left open up or we can't.

Case: 1 We can flip a card without opening new holes on the left.

In this case there must be more holes. Since this card's columns correspond to u_i and \bar{u}_i by flipping it we are in effect negating it in the truth assignment t . If by negating it we made all clauses true then C would be satisfiable, which is a contradiction. So there must be another clause that is false, and as shown above this means there is more holes in the K .

Therefore in this case there are still holes on the left column in K .

Case: 2 we cannot flip a card without opening new holes on the left.

In this case if we leave the card as is, then we haven't filled the hole on the left, and the TRUE card covers the right circles, which means we have a hole in the instance K . On the other hand if we flip the card, by assumption there will be a hole on the left by the assumption.

Therefore in this case there are still holes on the left column in K .

Therefore in all cases we found that regardless of our placement of clauses there will be a hole in the left column of K .

The only option at this point is to flip the TRUE card from covering the right holes to cover the left hole, however we can not do this because it will open holes on the right.

To prove this assume you can flip the TRUE card and there are no open holes on the right.

Then proceed with the following operation, we take all cards which are not the TRUE card and flip them so that now there are no holes on the left. We can then place the TRUE card to block the right holes. However this can not occur because we proved above, that no amount of flips of the cards would result in covering all holes on the left.

So there must be holes on the right, and if we flip the TRUE card to cover the holes on the left the holes on the right will go right through.

In either case we have a no instance K of Π .

Therefore if C is not satisfiable then K is a no instance of Π . \square

Therefore we have proven that C is satisfiable iff K is a yes instance of Π .

Therefore we have found that 3-SAT \propto Π . \square .