20 Mar 2019

1. Finish 3SAT $\leq_p$ IS
2. Some definitions & properties regarding NP-completeness
3. More hard problems!

Recall: ϕ is a 3CNF formula with n variables and ℓ clauses.

$$\phi = C_1 \wedge C_2 \wedge C_3 \wedge \cdots \wedge C_\ell$$

for all $i \in \{1,2,\ldots,\ell\}$ C_i is a Boolean OR of exactly 3 literals in the set $\{x_1,\ldots,x_n, \bar{x}_1,\ldots,\bar{x}_n\}$.

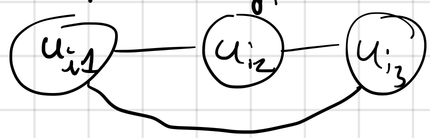
ϕ is satisfiable if $\exists$ a truth assignment to $x_1,\ldots,x_n$ such that each C_i contains at least one true literal.

Reduction $f: I_{3SAT} \to I_{IS}$.

$f(\phi)$ is the pair (G,k) where $k = n+\ell$ and G is a graph defined by following construction.

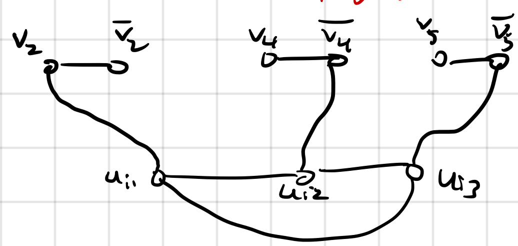
G has one vertex v_j or $\bar{v}_j$ for each literal x_j or $\bar{x}_j$.

G has edges $(v_j, \bar{v}_j)$ for all j.

G has a triangle  for each clause C_i.

G has edges connecting u_{ik} to $\bar{v}_j$ or v_j if $i \in \{1,\ldots,\ell\}$, $k \in \{1,2,3\}$, and x_j or $\bar{x}_j$ is the k^{th} literal in clause i. "pirate ship gadget"

E.g. $C_i = \bar{x}_2 \vee x_4 \vee x_5$



Reasoning: "Choosing a truth assignment of $x_1,\ldots,x_n$ is like choosing either v_j or $\bar{v}_j$ from each pair of nodes in the 'variable gadgets' part of the graph."

Choosing v_j represents setting $x_j = \text{TRUE}$

choosing $\bar{v}_j$ represents setting $x_j = \text{FALSE}$.

"Making three wrong choices should penalize the person searching for a large indep. set."

This happens because 3 truth assignments that fail to satisfy C_i correspond to 3 node choices that prevent choosing a node from the 'hull.'

Proving the reduction is correct.

Need to show:

(1) If ϕ is 'yes' instance of 3SAT (i.e. has a satisfying truth assignment) then G has an ind set with $n+\ell$ nodes.
"CONFIRM THAT THE GADGETS WORK AS INTENDED."

(2) If ϕ is a 'no' instance (i.e. has no satisfying assignment) then G has no ind set with $n+\ell$ nodes.
$\equiv$ If G has an ind set with $n+\ell$ nodes then ϕ has a satisfying truth assignment.
"CONFIRM THAT THE GADGETS HAVE NO UNINTENDED USES."

1. If $(x_1, \ldots, x_n)$ satisfies ϕ then define node set S as
- S contains $\begin{cases} v_j & \text{if } x_j = \text{TRUE} \\ \overline{v_j} & \text{if } x_j = \text{FALSE} \end{cases}$
- S contains u_{ik} if the k^{th} literal of clause i is satisfied and no lower numbered literal in the same clause is satisfied.

This forms an indep set because
- no "v-to-v" edges with both endpoints in S.
- no "u-to-u" edges " " " " ".
- no "u-to-v" edges " " " " "

because the nodes u_{ik} in S correspond to satisfied literals.

The size of S is $n+\ell$ because it contains one node per $\{v_j, \overline{v_j}\}$ (n of these) and one node per $\{u_{i1}, u_{i2}, u_{i3}\}$ (ℓ of these).

2. If $S \subset V(G)$ has $|S| = n+\ell$ and is indep set then set $x_j = \begin{cases} \text{TRUE} & \text{if } v_j \in S \\ \text{FALSE} & \text{if } v_j \notin S \end{cases}$.

Notice G has n "var gadgets", ℓ "clause gadgets" and an indep set contains ≤ 1 node per gadget.
Only way to have indep set of size $n+\ell$ is with exactly one node per gadget.
$\therefore$ $\forall j$ either S contains v_j and $x_j = \text{TRUE}$
or S contains $\overline{v_j}$ and $x_j = \text{FALSE}$.

Now each C_i is satisfied by $(x_1, \ldots, x_n)$: look at the literal corresponding to the node of C_i that is in S.

Remark. $\leq_p$ is a transitive relation.

If $A \leq_p B$ && $B \leq_p C$ it means $\exists$ functions f_{AB}, f_{BC} computable in poly time s.t.

$$\forall x \in I_A \qquad A(x) = B(f_{AB}(x))$$

$$\forall y \in I_B \qquad B(y) = C(f_{BC}(y))$$

$\Longrightarrow$ $\forall x \in I_A \quad A(x) = B(f_{AB}(x)) = C(f_{BC}(f_{AB}(x)))$

$\Longrightarrow$ $f_{BC} \circ f_{AB}$ is a (poly time) reduction from A to C.

Def. P is the set of decision problems solvable by poly-time alg.

NP is the set of decision problems that have a poly-time verifier: an algorithm V with two inputs x & y, called "problem instance" & "witness", s.t. V runs in poly-time in worst case and

$\forall x \quad A(x) = \text{yes} \Longleftrightarrow \exists y$ s.t. $|y| \leq \text{poly}(|x|)$ and $V(x,y) = \text{yes}$.

Observe the asymmetry: verifier for "yes" but not "no".