31 Mar 2021

More on NP Completeness

PRELIM 1 QUINTILES

<table>
<thead>
<tr>
<th>80%</th>
<th>46.5</th>
<th>45.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>45.0</td>
<td>43.0</td>
</tr>
<tr>
<td>40%</td>
<td>42.6</td>
<td>40.0</td>
</tr>
<tr>
<td>20%</td>
<td>36.0</td>
<td>37.4</td>
</tr>
</tbody>
</table>

Grades to be released on CMS after today's lecture.

Designing a reduction from 3SAT to INDEP SET.

That means: given a 3SAT instance (variables and clauses) we construct a graph $G$ and a number $k$ such that $G$ has a $k$-element independent set if and only if the given 3SAT instance is satisfiable.
Step 1. For each literal \( (x_i \text{ or } \overline{x_i}) \), create a vertex. For each clause, create 3 vertices.

Step 2. Join \( x_i \) to \( \overline{x_i} \) with edge. Join the 3 vertices of each “clause gadget” into a 3-cycle. Attach the vertices of this 3-cycle to the negations of the literals in that clause.

Step 3. Set \( k = (\# \text{ vars}) + (\# \text{ clauses}) \)

\[
(x_1 \lor x_2 \lor x_3) \land (\overline{x_2} \lor \overline{x_3} \lor \overline{x_4})
\]

\[ k = 4 + 2 = 6 \]

Running time of reduction: \( n \) vars, \( m \) clauses.
Step 1: $O(n+m)$
Step 2: $O(n+m)$
Step 3: $O(1)$

Correctness: $G$ has a $k$-element independent set if and only if the given 3SAT instance is satisfiable.

Show that given a satisfying truth assignment, we can get an independent set of $k$ elements.

The graph is composed of $K=n+m$ "gadgets."

A satisfying truth assignment gives us a way to choose one vertex from each gadget.

E.g., if $x_i = F$ and clause illustrated above contains $\bar{x}_i$, then no two elements of this set of $k$ vertices have an edge between them.
$(\Rightarrow)$ if the graph has an independent set of $k$ vertices, they must all belong to different gadgets. (every 2 vertices in same gadget are connected by an edge)

$\Rightarrow$ exactly one vertex in each gadget

$\Rightarrow$ let a truth assignment of $x_1, \ldots, x_n$ be defined by setting $x_i = T$ if node $x_i$ is in indep set

$x_i = F$ if node $\overline{x_i}$ is in indep set.

This truth assignment satisfies every clause b/c the node of the clause gadget that belongs to the indep set identifies a satisfied literal in that gadget.
If $A$ and $B$ are decision problems — computational problems with a Boolean answer (encoded as 1 for TRUE, 0 for FALSE) — then a poly-time reduction from $A$ to $B$ is an algorithm $R_{AB}$ running in poly time such that

\[ \forall x \ A(x) = B(R_{AB}(x)) \]

If such an alg exists we write $A \leq_p B$.

Note: This is a transitive relation.

If $R_{AB}$ reduces $A$ to $B$ and $R_{BC}$ reduces $B$ to $C$ then

\[ \forall x \ A(x) = B(R_{AB}(x)) = C(R_{BC}(R_{AB}(x))) \]

So $R_{AC} = R_{BC} \circ R_{AB}$ is a poly-time alg...
reducing A to C.

Interpretation: \( \leq_p \) ranks problems by computational difficulty.

Def: P is defined as a problem that can be solved in poly time.

NP is defined as decision problem A s.t. \( A \leq_p \text{3SAT} \).

A problem B is NP-Complete if \( B \leq_p \text{3SAT} \) and \( \text{3SAT} \leq_p B \).

Note that if \( \text{3SAT} \leq_p B \) then every problem in NP reduces to B.

\( A \in \text{NP} \) means \( A \leq_p \text{3SAT} \) then if \( \text{3SAT} \leq_p B \) we conclude \( A \leq_p B \).

To show a problem is NP-Complete we must:

1. show it belongs to NP (e.g. by reducing to 3SAT)
2. reduce any NP-Complete problem to it.
Conjecture: \( P \neq NP \)