# Homework 24

Brian Knotten, Brett Schreiber, Brian Falkenstein

October 23, 2018

## 15

Define the problem mentioned as $I - SAT$ (for inequality satisfiability). Show that CNFSAT $\leq_{poly}$ I-SAT. That is, show that we can construct an instance of I-SAT that is satisfiable if and only if the instance of CNF-SAT that we construct it from is also satisfiable.
For each literal $x$ in CNF-SAT, construct 2 inequalities as follows:

$$x + \overline{x} + x\overline{x} \leq 1$$

$$x + \overline{x} + x\overline{x} \geq 1$$

This forces every literal $x$ and its negation $\overline{x}$ to have one assigned to 1, and one to 0. If both were assigned to 0, it would violate the second inequality. If both were assigned to 1, it would violate the first inequality. If either were assigned to any value other than 0 or 1, it would violate one of the rules. Thus, every literal must be assigned a value of either 0 or 1, and both a literal and its negation may not have the same assignment. Take an assigned value of 1 to mean that literal is true in the CNF solution.
For each CNF-SAT clause, construct an inequality by adding the literals in the clause together, and setting it greater than or equal to 1. For example, for the clause $(x \vee \overline{y} \vee z)$:

$$x + \overline{y} + z \geq q$$

This forces at least one literal per clause to be true, while allowing for more than one to be. Doing this for every literal and clause will yield the I-SAT instance. Note that the transformation is in poly time with respect to the number of literals in the original CNFSAT instance (constant time to construct 2 inequalities for each literal) and the number of clauses (constant in the number of literals in each clause).

## 18

First, we will show that a similar problem that we are calling Fixed-Vertex Triad-Free Set is NP-Hard. Fixed Vertex Triad-Free Subset is a similar problem to finding a triad-free set in a graph $G$, except that one vertex $v \in G$ must be included in the solution set $S$.
Independent Set $\leq_p$ Fixed-Vertex Tried-Free Set. Given an input graph $G$ and an integer $k$, construct a graph $G' = G + v$, where $v$ is connected to every other vertex in $G'$. Call the Fixed Vertex Triad-Free Set algorithm on $G'$ and $k+1$, and let the fixed vertex be $v$. The algorithm will return true if and only if there exists an independent set of size $k$ in $G$.
This is because for every edge $(v_i, v_j) \in G$, there is a corresponding set of edges that forms a triangle $\{(v_i, v_j), (v_i, v), (v_j, v)\} \in G'$. Since $v$ must be in $S$, that means only one of $v_i$ and $v_j$ can be picked. If they were both picked, it would create a triad.
If $G$ has an independent set of size $k$, then $G'$ has a fixed-vertex triad-free set of size $k+1$. Add the fixed-vertex $v$ to $G$'s independent set to get a fixed-vertex triad-free set of size $k+1$ for $G'$. Conversely, if $G'$ has a fixed-vertex triad-free set of size $k+1$, then $G$ has an independent set of size $k$. Simply remove the fixed-vertex from $G''$s solution set to get an independent set of size $k$ for $G$.
Now we must show that Triad-Free Set is NP-Hard. Fixed-Vertex Triad-Free Set $\leq_p Triad-FreeSet$. The algorithm is as follows:

FixedVertexTriadFreeSetAlgorithm$(G, k, v)$:
    Let $S =$ TriadFreeSet$(G, k)$        if there does not exist a set, return false. It cannot be the case that there exists a solution to our stricter problem if there is not a solution to the more general problem.

If $v \in S$, return $S$, since the solution has $v$ in it.

Otherwise, $v \notin S$.

If $v$ is not a part of any triangles, then return a solution set $S'$ which includes $v$ and excludes some other arbitrary vertex. It is a valid solution since $v$ cannot inadvertently create any triads.

If $v$ is a part of some triangles, then...

## 20

In order to reduce the Vertex Cover problem to the presented problem, hereafter referred to as the Walkable Subset problem, given a the Graph $H$ and a desired cover size $\ell$, we will construct a graph $G$ such that $G$ has a walkable subset $WS$ of $G's$ vertices $V_G$ such that all the vertices in a given subset $R$ are within $WS$, $|WS| \leq$ and integer $k$, and, for every pair of vertices $x, y \in R$, one can walk from $x$ to $y$ in $G$ only traversing vertices in $WS$. We will construct $G$, $R$, and $k$ as follows:

Let $R =$ a single source vertex $v_s$ + one vertex $v_e$ for every edge in $H$
Let $G = R$ + one vertex $v_v$ for each vertex in $H$
We want $G$ to have edges so that the vertices in $R$ can be connected using only a few more vertices iff $H$ has a vertex cover of size $\ell$.
We accomplish this by connecting every vertex $v_v \in G$ corresponding to a vertex in $H$ to the source vertex $v_s$ and by connecting every $v_e$ to the vertices $v_v$ corresponding to the endpoints of the edge it is representing in $H$. Let $k = \ell + |E_H| + 1$ where $|E_H| =$ the number of edges in $H$ i.e. the number of $v_e$ vertices in $G$.

Clearly this transformation is polynomial in the size of $H$, as $G$ is constructed using one vertex for each edge ($|E_H|$) and vertex ($|V_H|$) in $H$ + one additional vertex ($v_s$) + $|E_H|$ edges to connect the $v_v$ to $v_s$ + $2|E_H|$ to connect every $v_e$ to its endpoints. So in total the transformation takes $\approx 4|E_H| + |V_H| + 1 = O(|E_H| + |V_H|)$.
To prove $G$ has a walkable subset including $R$ and at most $\ell + |E_H| + 1$ vertices iff $H$ has a vertex cover of size $\ell$ we must prove the relationship both ways:

First, assume $H$ has a vertex cover $VC$ of size $\ell$. A walkable subset $WS$ can be constructed from $G$ as follows: take the source vertex $v_s$, the edge vertices $v_e$, and $VC$. Note that there exists a path from $v_s$ to all $v_e$ by our construction of $G$ and the definition of $VC$ as a valid vertex cover. Further, $|WS| = |VC| + |E_H| + |v_s| = \ell + |E_H| + 1$ as required.

Second, assume $G$ has a walkable subset $WS$ of size at most $\ell + |E_H| + 1$. A vertex cover $VC$ of $H$ can be constructed simply by taking the vertices of $H$ that are vertices in $WS$ as no edge vertex $v_e$ of $WS$ has a single edge connected to the source vertex $v_s$ of $WS$, but must all be in $WS$ as $WS$ contains $R$ by definition, and $R$ contains all $v_e$ by our transformation. Therefore, $WS$ must have some vertices from $G$ such that there exists an edge from one of the vertices to all edge vertices i.e. the vertices form a vertex cover. This vertex cover is $\leq \ell$ as all $v_e$ and $v_s$ are in $WS$.
Therefore, if given a poly-time algorithm that solves Walkable Subset, we can solve Vertex Cover in polynomial time by using the poly-time transformation given above, a call to the poly algorithm, and outputting 1 if the poly-time algorithm does, and 0 otherwise.