CMPSCI 611: Advanced Algorithms

Lecture 18: NP-Completeness

Andrew McGregor

Last Compiled: November 13, 2009

Informal Summary from Last Time

1. Decision problem Π is in P if there is a polynomial time algorithm that correctly answers Π
2. Decision problem Π is in NP if there is a polynomial time algorithm that takes advice:
    - If the answer should be "yes", then there exists advice that leads the algorithm to output "yes"
    - If the answer is "no", then there doesn't exist advice that would lead the algorithm to output "yes"
3. A problem Π is NP-hard if for any $\Pi' \in NP$: $\Pi' \leq_P \Pi$
4. A problem Π is NP-complete if $\Pi \in NP$ and Π is NP-hard
5. To show Π is NP-complete it suffices to show that
    - Π is in NP
    - $\Pi' \leq_P \Pi$ for some Π' that is already known to be NP-complete
6. It's widely believed the $P \neq NP$ but finding a polynomial time algorithm for any NP-hard problem would prove $P = NP$.

Clique is NP-Complete

Theorem

Clique is NP-Complete

Proof.

1. We've already shown Clique $\in NP$
2. Because 3-SAT is NP-complete, it suffices to show 3-SAT $\leq_P$ Clique
3. Given formula 3-SAT

$$\phi = (l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge \ldots \wedge (l_{m,1} \vee l_{m,2} \vee l_{m,3})$$

in poly-time we can construct $G_\phi = (V_\phi, E_\phi)$:

$$V_\phi = \{l_{i,j} : i \in [m], j \in [3]\}$$

$$E_\phi = \{(l_{i,j}, l_{k,l}) : i, k \in [m], j \in [3], i \neq k, l_{i,j} \neq \overline{l}_{k,l}\}$$

4. Claim: ϕ is satisfiable iff G_ϕ has a clique of size m

□

ϕ is satisfiable iff G_ϕ has a clique of size m

Suppose ϕ is satisfiable:

1. In a satisfying assignment, at least one literal is true in each clause
2. Pick one true literal per clause: let Y be set of corresponding nodes
3. $G_\phi[Y]$ is a clique because $l_{i,j}$ and $\overline{l}_{k,l}$ can't both be in Y

Suppose G_ϕ has a clique of size m:

1. Let Y be the clique of size m
2. At most one node $l_{i,j}$ from i-th clause is in Y
3. Set $x_k = \text{TRUE}$ if $l_{i,j} = x_k$ and set $x_k = \text{FALSE}$ if $l_{i,j} = \bar{x}_k$

Vertex-Cover

Definition

A vertex cover of size k in graph $G = (V, E)$ is a subset $V' \subset V$ with

$$\forall (v, w) \in E : \text{either } u \in V' \text{ or } v \in V'$$

- Input: Given graph $G = (V, E)$ and integer k.
- Question: Does G contain a vertex cover of size k?

Vertex-Cover is NP-Complete

Theorem

Vertex-Cover is NP-Complete

Proof.

1. It is easy to show Vertex-Cover is in NP
2. It suffices to show 3-SAT $\leq_P$ Vertex-Cover
3. Given formula 3-SAT

   $$\phi = (l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge \ldots \wedge (l_{m,1} \vee l_{m,2} \vee l_{m,3})$$

   with n variables, in poly-time we can construct $G_\phi = (V_\phi, E_\phi)$:

   $$V_\phi = \{l_{i,j} : i \in [m], j \in [3]\} \cup \{T_i : i \in [n]\} \cup \{F_i : i \in [n]\}$$

   $$E_\phi = \begin{aligned} &\{(l_{i,j}, l_{i,l}) : i, k \in [m], j \in [3]\} \cup \{(T_i, F_i) : i \in [n]\} \\ &\cup \{(l_{i,j}, T_k) : l_{i,j} = x_k\} \cup \{(l_{i,j}, F_k) : l_{i,j} = \bar{x}_k\} \end{aligned}$$

4. Claim: ϕ is satisfiable iff G_ϕ has a vertex cover of size $n + 2m$

$\square$

ϕ is satisfiable impiles G_ϕ has a vertex cover of size $n + 2m$

Suppose ϕ is satisfiable:

1. Fix a satisfying assignment
2. Let $W = \{\, T_i : x_i = \text{TRUE}\} \cup \{\, T_i : x_i = \text{FALSE}\}$
3. Add l_{ij} to W if l_{ij} is there is no $w \in W$ with $(l_{i,j}, w) \in E_\phi$
4. W is a vertex cover of size $n + 2m$

G_ϕ has a vertex cover of size $n + 2m$ implies ϕ is satisfiable

Suppose G_ϕ has a vertex cover V' of size $n + 2m$:

1. For each $i \in [n]$, at least one of T_i or F_i must be in V'
2. For each $i \in [m]$, at least two of $l_{i,1}, l_{i,2}, l_{i,3}$ must be in V'
3. Since $|V'| = n + 2m$: "at least" must be "exactly"
4. Consider the assignment that sets $x_i = \text{TRUE}$ iff $T_i \in V'$
5. This satisfies ϕ because in the i-th clause there exists $l_{i,j} \notin V'$: if $l_{i,j} = x_i$ this means $T_i \in V'$ and if $l_{i,j} = \bar{x}_i$ then $F_i \in V'$.

Problem: Subset-Sum

- Input: A set S of n integers $\{s_1, s_2, \ldots, s_m\}$ and a target integer t.
- Question: Is there a subset $S' \subset S$ such that $t = \sum_{s \in S'} s$?

Subset-Sum is NP-Complete

Theorem

Subset-Sum is NP-Complete

Proof.

1. It is easy to show Subset-Sum is in NP
2. It suffices to show 3-SAT $\leq_P$ Subset-Sum
3. Given $\phi = (l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge (l_{2,1} \vee l_{2,2} \vee l_{2,3}) \wedge \ldots \wedge (l_{m,1} \vee l_{m,2} \vee l_{m,3})$ in n variables, define the set of integers (expressed in decimal):

   For $i \in [n]$: $s_i = (1, \underbrace{0, \ldots, 0}_{i-1}, y_m, \ldots, y_1)$, $s_i' = (1, \underbrace{0, \ldots, 0}_{i-1}, z_m, \ldots, z_1)$

   where $y_j = 1$ if x_i if a literal in j-th clause and 0 otherwise and $z_j = 1$ if $\bar{x}_i$ if a literal in j-th clause and 0 otherwise.

$$t = (\underbrace{1, \ldots, 1}_{n}, \underbrace{3, \ldots, 3}_{m})$$

$$h_1 = (1), h_2 = (1), h_3 = (1, 0), h_4 = (1, 0), \ldots, h_{2m} = (1, \underbrace{0, \ldots, 0})$$

ϕ is satisfiable implies there is a subset S' that sums to t

Suppose ϕ is satisfiable:

1. Fix a satisfying assignment
2. Let $S' = \{s_i : x_i = \text{TRUE}\} \cup \{s_i' : x_i = \text{FALSE}\}$
3. So far, for some $a_i \geq 1$:

$$\sum_{s \in S'} s = (\underbrace{1, \ldots, 1}_{n}, a_m, a_{m-1}, \ldots, a_1)$$

4. Can add "h" elements to S' such that

$$\sum_{s \in S'} s = (\underbrace{1, \ldots, 1}_{n}, \underbrace{3, \ldots, 3}_{m}) = t$$

Subset S' that sums to t implies ϕ is satisfiable

Suppose $\sum_{s \in S'} s = t$:

1. For each $i \in [n]$, exactly one of s_i and s_i' are in S'
2. Let x_i be TRUE is $s_i \in S'$ and FALSE otherwise
3. Since there are only two "h" elements corresponding to each clause, each clause most be satisfied.

For Next Time...

- Finish reading Chapter 7.