CSCI 1590
Intro to Computational Complexity

NP-Complete Languages

John E. Savage

Brown University

February 8, 2008

1. Definition of **NP**-Complete Languages

2. VERTEX COVER is **NP**-complete

3. SUBSET SUM is **NP**-complete

4. PARTITION is **NP**-complete

Definition of **NP**-complete Languages

Definition

A language L_2 is **NP**-complete if a) L_2 is in **NP** and b) for every language L_1 in **NP** there is polynomial-time reduction from it to L_2 (L_2 is **NP-hard**.)

$f : \Sigma_1^* \mapsto \Sigma_2^*$ is a **reduction** from L_1 to L_2 if $x \in L_1$ if and only if $f(x) \in L_2$. f is a **polynomial-time reduction** if it can also be computed by a DTM in polynomial time in the length of its input.

A language L_1 is in **NP** if there is an NTM M_1 that accepts every $\mathbf{x} \in L_1$ in $p(n)$ steps where $n = |\mathbf{x}|$ and $p(n)$ is a polynomial and does not accept any $\mathbf{x} \notin L_1$.

Three **NP**-Complete Languages

SAT

Instance: Literals $X = \{x_1, \overline{x}_1, x_2, \overline{x}_2, \ldots, x_n, \overline{x}_n\}$, and clauses $C = (c_1, c_2, \ldots, c_m)$ where each clause c_i is a subset of X.
Answer: "Yes" if for some assignment of Booleans to variables in $\{x_1, x_2, \ldots, x_n\}$, at least one literal in each clause has value 1.

3-SAT

Instance: SAT instance which has at most three literals/clause.
Answer: "Yes" if for some assignment of Booleans to variables in $\{x_1, x_2, \ldots, x_n\}$, at least one literal in each clause has value 1.

NAESAT

Instance: An instance of 3-SAT.
Answer: "Yes" if each clause is satisfiable when not all literals have the same value.

Proof that VERTEX COVER is **NP**-complete

VERTEX COVER
Instance: A graph $G = (V, E)$ and an integer k.
Answer: "Yes" if there is a set of k vertices such touching each edge.

Theorem

VERTEX COVER *is* **NP**-*complete.*

Proof

Clearly each "Yes" instance of VERTEX COVER is in **NP**. We reduce INDEPENDENT SET to it. Given an instance $IS = (G, k)$ of INDEPENDENT SET where $G = (V, E)$, produce an instance $VC = (G, (n - k))$ $(n = |V|)$ of VERTEX COVER.

Proof that VERTEX COVER is **NP**-complete

Proof (cont.)

If IS is a “Yes” instance of INDEPENDENT SET, there is a set $S \subseteq V$, $|S| = k$, such that no two vertices in S are adjacent. In VC, the set $T = V - S$, $|T| = (n - k)$, is a vertex cover. This follows because edges adjacent to vertices in S are covered by vertices in T as well as all other edges.

Similarly, if VC is a “Yes” instance of of VERTEX COVER, there is a $T \subseteq V$, $|T| = n - k$, such that every edge in G is adjacent to a vertex in T. The vertices in $V - T$, $|V - T| = k$, are independent because if they were adjacent, at least one of them would be in T.

It follows that an instance of INDEPENDENT SET is a “Yes” instance if and only if the corresponding instance of VERTEX COVER is a “Yes” instance.

Proof that SUBSET SUM is **NP**-complete

SUBSET SUM
Instance: A set $S = \{b_1, b_2, \dots, b_n\}$ of n integers and an integer t.
Answer: "Yes" if a subset of S adds to t.

If an instance of SUBSET SUM is a "Yes" instance, there exists a subset $I \subseteq S$ such that $\sum_{i \in I} b_i = t$.

Proof that SUBSET SUM is **NP**-complete

Theorem

SUBSET SUM *is* **NP***-complete.*

Proof

Given instance $C = (c_1, c_2, \ldots, c_m)$ of 3-SAT with literals $X = \{x_1, \overline{x}_1, x_2, \overline{x}_2, \ldots, x_n, \overline{x}_n\}$, construct instance of SUBSET SUM containing $2n + 3m + 1$ integers (in decimal) grouped into four sets, $V = \{v_1, \bar{v}_1, v_2, \bar{v}_2, \ldots, v_n \bar{v}_n\}$, $S_0 = \{s_{1,0}, s_{2,0}, \ldots, s_{m,0}\}$, $S_1 = \{s_{1,1}, s_{2,1}, \ldots, s_{m,1}\}$, $S_2 = \{s_{1,2}, s_{2,2}, \ldots, s_{m,2}\}$, and the single integer t which is $1^n 3^m 1^m$ in decimal.

Proof that SUBSET SUM is NP-complete

Proof (cont.)

Let v_i be the decimal integer $v_i = 0^{i-1}10^{n-i}\sigma_i 0^m$ where $\sigma_i \in \{0,1\}^m$ has a 1 in the jth position if x_i is a literal in clause c_j. $\bar{v}_i$ is defined in the same way except for $\bar{x}_i$ instead of x_i. See below for clauses $c_1 = (x_1 \vee \bar{x}_3)$ and $c_2 = (\bar{x}_1 \vee x_2 \vee x_3)$.

VarName	p_1	p_2	p_3	q_1	q_2	r_1	r_2
$v_1 =$	1	0	0	1	0	0	0
$\bar{v}_1 =$	1	0	0	0	1	0	0
$v_2 =$	0	1	0	0	1	0	0
$\bar{v}_2 =$	0	1	0	0	0	0	0
$v_3 =$	0	0	1	0	1	0	0
$\bar{v}_3 =$	0	0	1	1	0	0	0

Proof that SUBSET SUM is **NP**-complete

Proof (cont.)

For $b = 0, 1, 2$ and $1 \le j \le m$ let $s_{j,b} = 0^n 0^{j-1} b 0^{m-j} 0^{j-1} 1 0^{m-j}$. These are **slack variables**. Recall that $t = 1^n 3^m 1^m$. See example.

VarName	p_1	p_2	p_3	q_1	q_2	r_1	r_2
$s_{1,0} =$	0	0	0	0	0	1	0
$s_{1,1} =$	0	0	0	1	0	1	0
$s_{1,2} =$	0	0	0	2	0	1	0
$s_{2,0} =$	0	0	0	0	0	0	1
$s_{2,1} =$	0	0	0	0	1	0	1
$s_{2,2} =$	0	0	0	0	2	0	1
$t =$	1	1	1	3	3	1	1

A subset sums to t if only if for each $1 \le i \le n$ either $x_i = 1$ $\bar{x}_i = 1$ and due to column r_j one of $s_{j,0}, s_{j,1}, s_{j,2}$ is used. v_i's and $\bar{v}_i$'s in column q_j add to 0, 1, or 2. Choose slacks variables so sum is t.

Proof that PARTITION is **NP**-complete

PARTITION
Instance: A set $C = \{c_1, c_2, \ldots, c_n\}$ of integers.
Answer: "Yes" if S can be partitioned into two disjoint sets U and V such that the sum of the integers in U is equal to the sum in V.

Theorem

PARTITION *is* **NP***-complete.*

Proof

Clearly PARTITION is in **NP**. To show it is **NP**-hard, reduce an instance SS of SUBSET SUM (a set $S = \{b_1, b_2, \ldots, b_n\}$ of integers and an integer t) to an instance P of PARTITION ($C = \{b_1, b_2, \ldots, b_n, c, d\}$ where $c = 2A - t$, $d = A + t$ and $A = \sum_j b_j$). The sum of elements in C is $4A$. C is a "Yes" instance of PARTITION if and only if there are disjoint sets $U \subseteq C$ and $V \subseteq C$ such that their sums are each $2A$.

Proof that PARTITION is **NP**-complete

Proof (cont.)

Let SS be a "Yes" instance of SUBSET SUM. Then there is a set I such that $\sum_{i \in I} b_i = t$. Adding c to this set produces a "Yes" instance of PARTITION. On the other hand, suppose that P is a "Yes" instance of PARTITION. Because neither U nor V can contain both c and d, one of the two, say U, contains c. The remaining elements in U must sum to t. This subset forms a "Yes" instance of SUBSET SUM.

Thus, SS is a "Yes" instance of SUBSET SUM if and only if P is a "Yes" instance of PARTITION.