Reductions-2

Theorem 3 (Cook-Levine) *SAT is $\mathcal{NP}$–complete.*

SAT is the first problem proved $\mathcal{NP}$–complete. Additional problems have been proved $\mathcal{NP}$–complete by reduction from SAT, or other problems already known to be $\mathcal{NP}$–complete, using Theorem 4 below.

Theorem 4 *If L_1 is NP–complete and L_2 is $\mathcal{NP}$, and there is a polynomial reduction of L_1 to L_2 , then L_2 is $\mathcal{NP}$–complete.*

Proof: Condition 1 of the definition of $\mathcal{NP}$–completeness is given. To prove Condition 2, let $L_3 \in \mathcal{NP}$. We need to show that there is a polynomial time reduction R of L_3 to L_2. Since L_1 is $\mathcal{NP}$–complete, there is a polynomial time reduction of L_3 to L_1, and we are given a polynomial time reduction of L_1 to L_2. Let R be the composition of those two reductions. □

Theorem 3 is the Cook-Levine theorem. The proof is available in various sources, including the internet.

For any $k \geq 2$, we define k-SAT to be the satisfiability problem for Boolean expressions in K-CNF form, meaning in CNF form where each clause has k terms.[1]

Theorem 5 *For any $k \geq 3$, k-*SAT *is $\mathcal{NP}$–complete.*

Proof: We give a polynomial time reduction of SAT to k-SAT.

This proof is not finished.

The result follows from Theorems 3 and 4. □

We remark that 2-SAT is polynomial.

Let IND be the independent set problem: given a graph G and an integer k, does G have an independent set of order k? A set of vertices I of G is *independent* if no two members of I are neighbors.

Theorem 6 IND *is $\mathcal{NP}$–complete.*

Proof: We give a polynomial time reduction R of 3-SAT to IND.

Let e be a Boolean expression in 3-CNF form, the conjunction of k clauses, each with three terms. Then $e = C_1 * C_2 * \cdots * C_k$ where each clause $C_i = t_{i,1} + t_{i,2} + t_{i,3}$ and each $t_{i,j}$ is either a variable or the negation of a variable. Let $R(e) = G$, where $G =$

[1] We allow a clause to have fewer than k terms, since we can pad the clause with duplicate terms. For example, we could replace the clause $x+!y$ by the equivalent $x + x+!y$.

(V, E), a graph of $3k$ vertices $\{v[i,j] : 1 \leq i \leq k,\, j = 1,2,3\}$, and E is the set of pairs $\{\,\{v[t_{i,j}], v[t_{i',j'}]\} : i = i' \text{ or } t_{i,j} * t_{i',j'} \text{ is a contradiction.}\}$. We call an edge $\{\,\{t_{i,j}, t_i, j'\}\}$ em short, and the other edges *long*.

We now show that R is a reduction of 3-SAT to IND. Suppose IND has a set I of k independent vertices. We define an assignment for each variable in e as follows. For $v[i,j] \in I$, either $t_{i,j}$ is either x or $!x$ for some variable x. If it is x, we assign x true, otherwise false. If any variable of e is not yet assigned, assign it arbitrarily to true.

A variable x cannot be assigned both true and false. If $t[i,j] = x$ and $t[i',j'] = !x$, then there is a long edge between $v[i,j]$ and $v[i',j']$, and hence those two vertices cannot both be members of I.

Since no two of members of I can be connected by a short edge, I contains exactly one vertex $v[i,j]$ for each i, hence one term of C_i is true. Thus, each clause is true, hence e is true.

Conversely, assume that e has a satisfying assignment. For each clause C_i, choose one term t_{i,j_i} which is true under the assignment. Then $I = \left\{v_[i, j_i]\right\}$ is an independent set of G, since no two of those terms contradict, and hence there is no long edge connecting them and, Since there is one vertex of each i in I, no two are connected by a short edge.

The result follows from Theorems 5 and 4. $\square$

The subset sum problem, which we abreviate a SSP is whether, given a set of weights, there is a subset whose total weight is equal to a given constant. More formally, an instance of SSP is a pair (σ, K) where K is a constant and $\sigma = x_1, x_2, \ldots x_n$, a sequence of numbers. We will use the variant of SSP where all numbers are positive. The problem is, does σ have a subsequence whose total is K?

Theorem 7 SSP *is $\mathcal{NP}$–complete.*

Proof: We give a polynomial time reduction R of IND to SSP.

This proof is not finished.

The result follows from Theorems 6 and 4. $\square$

An instance of Partition is a pair (σ, K) where σ is a sequence of numbers and K is a number. A solution to that instance is a subsequence of σ whose total is half the total of σ. We use the variant of Partition where the numbers are positive.

Theorem 8 Partition *is $\mathcal{NP}$–complete.*

Proof: We give a polynomial time reduction R of SSP to Partition. Let (σ, K) be an instance of SSP. Let $\sigma = x_1, x_2, \ldots, x_n$. Let $S = \sum_{i=1}^{n} x_i$. Without loss of generality, $K \leq S$, since otherwise there can be no solution.

We define $R(\sigma, K)$ to be a sequence τ obtained by appending two more terms to σ: $\tau = x_1, x_2, \ldots, x_n, x_{n+1}, x_{n+2}$, where $x_{n+1} = K + 1$ and $x_{n+2} = S - K + 1$. The sum of the terms of τ is $2S + 2$, and thus a solution is a subsequence of τ whose total is $S + 1$. If there is a subsequence σ' of σ whose total is K, the subsequence of τ obtained by appending x_{n+2} to σ' has total $S + 1$.

Conversely, suppose τ has a subsequence τ' of total $S + 1$. τ' cannot contain both x_{n+1} and x_{n+2}, since their total is greater than $S + 1$. Similarly, τ' must contain at least of those terms, since otherwise any subsequence would have total less than $S + 1$.

If τ' contains x_{n+2}, the remaining terms of τ' are a subsequence of σ whose total is K. Otherwise, the subsequence consisting of those terms of τ not in τ' also has total $S + 1$, and those terms of that subsequence, after x_{n+2} is deleted, total K.

The result follows from Theorems 7 and 4. $\square$