LECTURE 11 – REDUCTIONS AND COMPLETENESS

Overview

- Proving that 3-SAT is NP-complete
- Proving that INDEPENDENT SET in NP-complete

Proving that 3-SAT is NP-Complete

The SAT problem: given U, a finite set of Boolean variables and C, a finite set of clauses over U, is there an assignment of values to the set of variables U that satisfies the set of clauses C to be true?

$$(q_{1,1} \vee q_{1,2} \vee \ldots \vee q_{1,m_1}) \wedge (q_{2,1} \vee q_{2,2} \vee \ldots \vee q_{2,m_2}) \wedge \ldots \wedge (q_{n,1} \vee q_{n,2} \vee \ldots \vee q_{n,m_n})$$

Example of an instance of SAT with |U|=m=4 variables and |C|=n=4 clauses:

$$(u_1 \vee u_2 \vee u_3 \vee \overline{u}_4) \wedge (\overline{u}_2) \wedge (\overline{u}_1 \vee u_2 \vee \overline{u}_3) \wedge (u_1 \vee \overline{u}_2)$$

The 3-SAT problem: given U, a finite set of Boolean variables and C, a finite set of clauses over U, where each clause consists of 3 variable. Is there an assignment of values to the set of variables U that satisfies the set of clauses C to be true?

$$(q_{1,1} \vee q_{1,2} \vee q_{1,3_1}) \wedge (q_{2,1} \vee q_{2,2} \vee q_{2,3_2}) \wedge \ldots \wedge (q_{n,1} \vee q_{n,2} \vee q_{n,3_n})$$

Example of an instance of 3-SAT with |U|=m=4 variables and |C|=n=4 clauses:

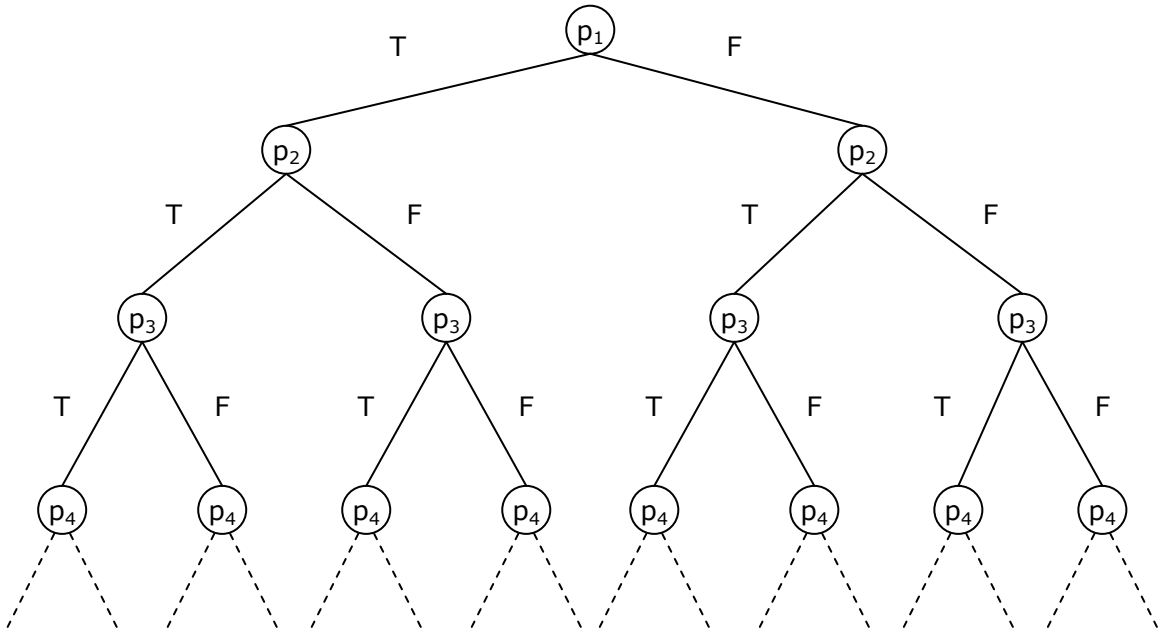
$$(u_1 \vee u_2 \vee u_3) \wedge (\overline{u}_2 \vee \overline{u}_3 \vee u_4) \wedge (\overline{u}_1 \vee u_2 \vee \overline{u}_4) \wedge (u_1 \vee \overline{u}_2 \vee u_3)$$

Why 3-SAT is Hard?

3-SAT is verifiable in polynomial time, that is, it is easy to verify whether a set of clauses each of them with 3 literals is true or not.

Intuitively, the reason 3-SAT is hard is that given a set of clauses with 3 literals over a set of Boolean variables, the Turing machine must make choices of truth values T or F for each variable.

For a 3-SAT instance with variables p_1, p_2,…,p_m and n clauses, the corresponding tree is shown below. For each choice (branch in the tree) a Turing machine can easily check the truth value of the n clauses using the usual truth Table method.



3-SAT belongs to the class NP because given the Boolean expression made of n clauses, it is easy to verify if an assignment of values to the m Boolean variables satisfies the expression.

The formal proof will be based on reducing SAT to 3-SAT, that is, SAT $\propto$ 3-SAT.

Given U={u_1,u_2,…,u_m} the set of Boolean variables and C={c_1,c_2,c_3,…,c_n} the set of clauses,

Construct a collection C' of 3-literal clauses on a set of U' variables such that C' is satisfiable only of C is satisfiable.

Then, any instance of SAT can be transformed to an instance of 3-SAT and since SAT is known to be NP-complete, then 3-SAT will be proven to be NP-complete as well.

Replace each clause c_j in C by an equivalent set of clauses c_j' with 3-literals based on the variables in U and some additional variables U_j'.

Then $U' = U \cup \left(\bigcup_{j=1}^{n} U_j' \right)$ and $C' = \bigcup_{j=1}^{n} c_j$

The construction of c_j' and U_j' is as follows:

Assume c_i is made with the literals $\{z_1,z_2,\ldots,z_k\}$ which are literals made from the variables in U.

If k=1 then

$U_j' = \{y_j^1, y_j^2\}$

$c_j' = (z_1 \vee y_j^1 \vee y_j^2) \wedge (z_1 \vee y_j^1 \vee \overline{y}_j^2) \wedge (z_1 \vee \overline{y}_j^1 \vee y_j^2) \wedge (z_1 \vee \overline{y}_j^1 \vee \overline{y}_j^2)$

If k=2 then

$U_j' = \{y_j^1\}$

$c_j' = (z_1 \vee z_2 \vee y_j^1) \wedge (z_1 \vee z_2 \vee \overline{y}_j^1)$

If k=3 then

$U_j' = \phi$

$c_j' = (c_j)$

If k>3 then

$U_j' = \{y_j^i : 1 \le i \le k-3\}$

$c_j' = (z_1 \vee z_2 \vee y_j^1) \cup \{(\overline{y}_j^i, z_{i+2}, y_j^{i+1}) : 1 \le i \le k-4\} \cup \{(\overline{y}_j^{k-3}, z_{k-1}, z_k)\}$

Illustrating the transformation from a SAT instance to a 3-SAT instance:

$$(u_1 \vee u_2 \vee u_3 \vee \overline{u}_4) \wedge (\overline{u}_2) \wedge (\overline{u}_1 \vee u_2 \vee \overline{u}_3) \wedge (u_1 \vee \overline{u}_2)$$

$(\overline{u}_2) \Rightarrow (\overline{u}_2 \vee p \vee q) \wedge (\overline{u}_2 \vee \overline{p} \vee q) \wedge (\overline{u}_2 \vee p \vee \overline{q}) \wedge (\overline{u}_2 \vee \overline{p} \vee \overline{q}) = c_2$

$(u_1 \vee \overline{u}_2) \Rightarrow (u_1 \vee \overline{u}_2 \vee r) \wedge (u_1 \vee \overline{u}_2 \vee \overline{r}) = c_4$

$(\overline{u}_1 \vee u_2 \vee \overline{u}_3) \Rightarrow (\overline{u}_1 \vee u_2 \vee \overline{u}_3) = c_3$

$(u_1 \vee u_2 \vee u_3 \vee \overline{u}_4) \Rightarrow (u_1 \vee u_2 \vee w_1) \wedge (\overline{w}_1 \vee u_3 \vee \overline{u}_4) = c_1$

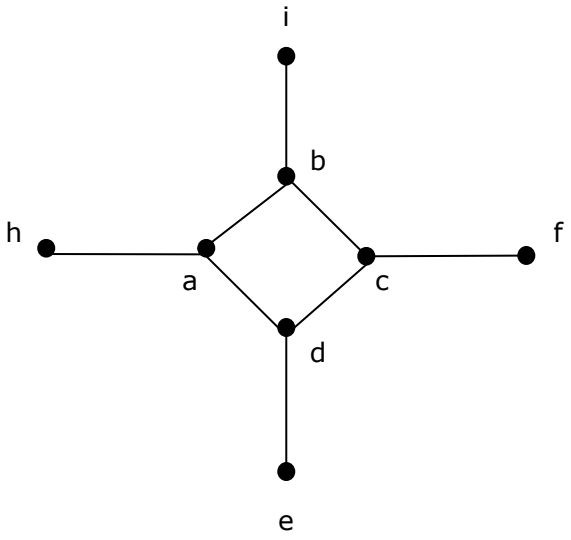
The 3-SAT instance which is made of the set of clauses $\{c_1,c_2,c_3,c_4\}$ is satisfiable if and only if the original SAT instance is satisfiable.

The reduction can be done in polynomial time because the number of literals is C' is bounded by a polynomial in n·m.

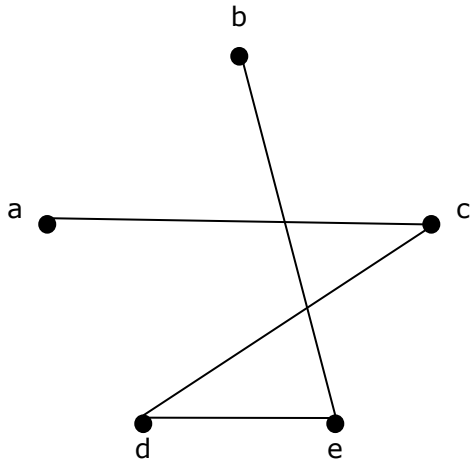
Proving that INDEPENDENT SET is NP-Complete

The INDEPENDENT SET problem: given a graph G=(V,E) and a positive integer bound k, is there an independent set (for every pair of vertices i,j in an independent set, there is no edge between i and j) of vertices $I \subseteq V$ with |I|=k?

Example of an instance of INDEPENDENT SET, in the following graph I={e,f,h,i} is and independent set of size 4, I={a,i,c} is an independent set of size 3 and I={b,d} is an independent set of size 2. What is maximal independent set?



What is the maximal independent set in the following graph?

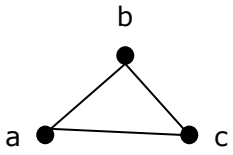


INDEPENDENT SET belongs to the class NP because given the graph G and the set I it is easy to verify if there are edges or not between all the vertices in I.

The formal proof will be based on reducing 3-SAT to INDEPENDENT SET, that is, 3-SAT $\propto$ INDEPENDENT SET.

For this, it should be noted that in a 'triangle' there is no INDEPENDENT SET.

A 'triangle' is a complete graph with 3 vertices:



Given a 3-SAT instance, the set of Boolean variables $U=\{u_1,u_2,\ldots,u_m\}$ and the set of clauses with 3 literals $C=\{c_1,c_2,c_3,\ldots,c_n\}$,

For each of the n clauses construct a separate 'triangle' where each vertex in the 'triangle' corresponds to a literal in the clause.

We add an edge between two vertices in different triangles if and only if the vertices correspond to complementary literals (e.g. u_1 and $\overline{u}_1$).

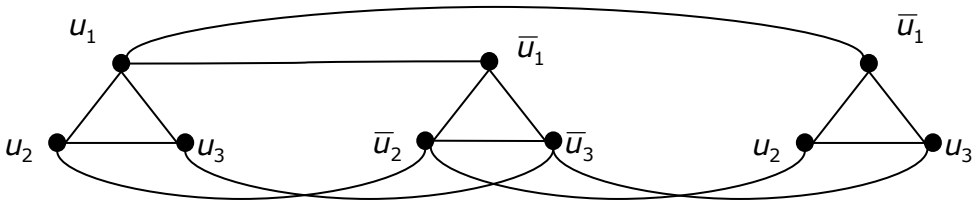
Then, any instance of 3-SAT can be transformed to an instance of INDEPENDENT SET and since 3-SAT is known to be NP-complete, then INDEPENDENT SET will also be proven to be NP-complete.

Illustrating the transformation from a 3-SAT instance to an INDEPENDENT SET instance:

The 3-SAT instance:

$$(u_1 \vee u_2 \vee u_3) \wedge (\overline{u}_1 \vee \overline{u}_2 \vee \overline{u}_3) \wedge (\overline{u}_1 \vee u_2 \vee u_3)$$

is transformed to the graph G:



There is an INDEPENDENT SET of k vertices (the number of triangles) in G if and only if the 3-SAT Boolean expression with n (number of clauses) is satisfiable.

Suppose that an independent set I exists, since k=n, I must only contain a vertex from each triangle.

Since the vertices are labelled with literals and I contains no two vertices corresponding to complementary literals, I is a true assignment that satisfies the Boolean expression.

The true literals are just those which are labels of vertices in I and variables left unassigned can take any value.

This is a true assignment because any two complementary literals are connected by and edge in G and so they cannot be both in I.

Conversely, if the Boolean expression is satisfiable then we can identify a true literal in each clause and the vertex labelled with that literal in that clause is added to I until we have k vertices.

The reduction can be done in polynomial time because the number of vertices in G is n·m while the number of edges in G is bounded by n·m.

Reading: (Garey&Johnson, chapter 3)