CURSO DE COMPLEJIDAD COMPUTACIONAL 2021

Práctico 2

Temas : Clases P, NP y coNP. Reducción polinomial de Karp en la clase NP. Problemas NP-completos.

2.1 Sipser

A ***2cnf-formula*** is an AND of clauses, where each clause is an OR of at most two literals. Let $2SAT = \{\langle\phi\rangle \mid \phi \text{ is a satisfiable 2cnf-formula}\}$. Show that $2SAT \in \mathrm{P}$.

2.2 Sipser

Call graphs G and H ***isomorphic*** if the nodes of G may be reordered so that it is identical to H. Let $ISO = \{\langle G, H\rangle \mid G \text{ and } H \text{ are isomorphic graphs}\}$. Show that $ISO \in \mathrm{NP}$.

2.3 Ejercicio 2.8 Arora -Barak

Let HALT be the Halting language defined in Theorem 1.11. Show that HALT is **NP**-hard. Is it **NP**-complete?

2.4 Sipser.

Let G represent an undirected graph. Also let

$$SPATH = \{\langle G, a, b, k\rangle \mid G \text{ contains a simple path of length at most } k \text{ from } a \text{ to } b\},$$

and

$$LPATH = \{\langle G, a, b, k\rangle \mid G \text{ contains a simple path of length at least } k \text{ from } a \text{ to } b\}.$$

a. Show that $SPATH \in \mathrm{P}$.

b. Show that $LPATH$ is NP-complete. You may assume the NP-completeness of $UHAMPATH$, the Hamiltonian path problem for undirected graphs.

2.5 (Sipser) Explicar porqué generalmente se cree que *PATH* no es NP-completo. Mostrar que probar que *PATH* NO es NP-completo equivale a probar que $P \neq NP$.

2.6 El problema de Ubicación Optima de Plantas industriales ***UOP*** toma como entrada un grafo no dirigido con un entero positivo asociado a cada vértice (peso del vértice) y un entero positivo t. Los vértices representan posibles ubicaciones para las plantas, el peso de cada vértice mide el valor que tiene una planta en esa ubicación particular y una arista (u, v) significa que los vértices u y v no pueden tener *ambos* una planta asignada (no es admisible que coexistan dos plantas en esas locaciones). La salida del problema es un booleano que indica si es posible localizar plantas con un valor total de *al menos* t sin que haya conflictos.

Probar que ***UOP*** es NP-completo.

2.7 Una tienda que trata de analizar el comportamiento de sus clientes a menudo mantiene una tabla bidimiensional A, donde las filas corresponden a sus clientes y las columnas a los productos que vende. El elemento $A(i, j)$ especifica la cantidad del producto j que ha sido comprada por el cliente i. Un ejemplo de tabla se muestra a continuación.

Cliente	Detergente	Cerveza	Pañales	Arena para gatos
Raj	0	6	0	3
Alanis	2	3	0	0
Chelsea	0	0	0	7

Se dice que un subconjunto *S* de clientes es ***diverso*** si ningún par de clientes en *S* ha comprado nunca el mismo producto, es decir que para cada producto a lo sumo un cliente de *S* lo ha comprado alguna vez. Un conjunto diverso de clientes resulta útil por ejemplo como grupo objetivo para estudios de mercado. En la tabla ejemplo el conjunto de clientes {Alanis, Chelsea} es diverso.

Se define el problema de Conjunto Diverso como sigue: dada una tabla $A_{m \times n}$ con m clientes y n productos y un número $k \leq m$, ¿existe un subconjunto de al menos k clientes que sea diverso?

Probar que Conjunto Diverso es NP-completo.

2.8 Sea

$$\text{ATLEAST2-SAT} = \{\phi \mid \phi \text{ es una fórmula } 3CNF \text{ con al menos 2 asignaciones} \neq \text{que la satisfacen}\}$$

Probar que el problema ATLEAST2-SAT es NP-completo.

2.9 Ejercicio 2.15 Arora -Barak

In the CLIQUE problem, we are given an undirected graph G and an integer K and have to decide whether there is a subset S of at least K vertices such that every two distinct vertices $u, v \in S$ have an edge between them (such a subset is called a *clique* of G). In the VERTEX COVER problem, we are given an undirected graph G and an integer K and have to decide whether there is a subset S of at most K vertices such that for every edge $\overline{i\,j}$ of G, at least one of i or j is in S (such a subset is called a *vertex cover* of G). Prove that both these problems are **NP**-complete.

In both cases reduce from problem INDSET, assumed to be NP-complete.

2.10 Sipser

A subset of the nodes of a graph G is a ***dominating set*** if every other node of G is adjacent to some node in the subset. Let

$$DOMINATING\ SET = \{\langle G, k\rangle / G \textit{ has a dominating set with at most } k \textit{ nodes}\}$$

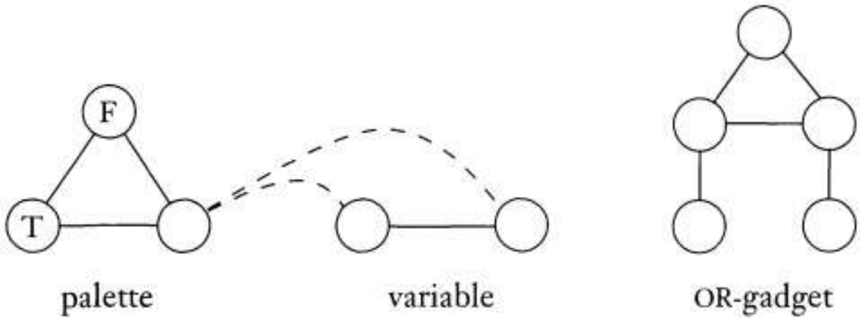
Show that it is NP-complete by giving a reduction from *VERTEX-COVER*.

2.11 Sipser

A ***coloring*** of a graph is an assignment of colors to its nodes so that no two adjacent nodes are assigned the same color. Let

$$3COLOR = \{\langle G\rangle | \text{ the nodes of } G \text{ can be colored with three colors such that no two nodes joined by an edge have the same color}\}.$$

Show that *3COLOR* is NP-complete. (Hint: Use the following three subgraphs.)



2.12 Sipser

Let ϕ be a 3cnf-formula. An ***$\neq$-assignment*** to the variables of ϕ is one where each clause contains two literals with unequal truth values. In other words, an $\neq$-assignment satisfies ϕ without assigning three true literals in any clause.

a. Show that the negation of any $\neq$-assignment to ϕ is also an $\neq$-assignment.

b. Let $\neq SAT$ be the collection of 3cnf-formulas that have an $\neq$-assignment. Show that we obtain a polynomial time reduction from *3SAT* to $\neq SAT$ by replacing each clause c_i

$$(y_1 \vee y_2 \vee y_3)$$

with the two clauses

$$(y_1 \vee y_2 \vee z_i) \quad \text{and} \quad (\overline{z_i} \vee y_3 \vee b),$$

where z_i is a new variable for each clause c_i and b is a single additional new variable.

c. Conclude that $\neq SAT$ is NP-complete.

The problem $\neq SAT$ is also known as NAE 3SAT (Not All Equal 3SAT).

2.13 Probar que un lenguaje L es NP-completo si y solo si $\bar{L}$ es coNP-completo.

2.14 Ejercicio 2.15 Arora -Barak

Prove that Definitions 2.19 and 2.20 do indeed define the same class **coNP**.