5.3. Reducciones para NP-completitud

Definición. Una fórmula está en CNF si es una conjunción de disyunciones de literales.

Definición (SAT-CNF)**. Input:** Una fórmula en *CNF*, $\varphi = C_1 \wedge \cdots \wedge C_m$.
Output: ¿Hay una asignación que satisfaga φ?

$$\text{SAT-CNF} = \{\langle\varphi\rangle : \varphi \text{ es una CNF satisfacible}\}.$$

Teorema. SAT-CNF es NP-completo.

Definición (3SAT)**. Input:** Una fórmula en *3CNF*, $\varphi = C_1 \wedge \cdots \wedge C_m$, con tres literales de variables distintas por cláusula.
Output: ¿Hay una asignación que satisfaga φ?

$$\text{3SAT} = \{\langle\varphi\rangle : \varphi \text{ es una 3CNF satisfacible}\}.$$

Teorema. 3SAT es NP-completo.

Definición (Independent Set)**. Input:** Un grafo G y un número k.
Output: ¿Hay al menos k vértices sin aristas entre ellos?

$$\text{IS} = \{\langle G, k\rangle : G \text{ tiene un conjunto independiente de tamaño } \geq k\}.$$

Teorema. IS es NP-completo.

Definición (Clique)**. Input:** Un grafo G y un número k.
Output: ¿Hay al menos k vértices todos adyacentes entre sí?

$$\text{CLIQUE} = \{\langle G, k\rangle : G \text{ tiene un clique de tamaño } \geq k\}.$$

Teorema. CLIQUE es NP-completo.

Definición (Vertex Cover)**. Input:** Un grafo G y un número k.
Output: ¿Hay a lo más k vértices que toquen todas las aristas?

$$\text{VC} = \{\langle G, k\rangle : G \text{ tiene una cubrimiento por vértices de tamaño } \leq k\}.$$

Teorema. VC es NP-completo.

Definición (Set Cover)**. Input:** Un conjunto U, una colección $\mathcal{S}$ de subconjuntos de U y un número k.
Output: ¿Hay a lo más k conjuntos de $\mathcal{S}$ cuya unión sea U?

$$\text{SC} = \{\langle U, \mathcal{S}, k\rangle : \exists \mathcal{C} \subseteq \mathcal{S},\ |\mathcal{C}| \leq k,\ \bigcup \mathcal{C} = U\}.$$

Teorema. SC es NP-completo.

Definición (Hamiltonian Cycle)**. Input:** Un grafo dirigido G.
Output: ¿Hay un ciclo que visite cada vértice exactamente una vez?

$$\text{HAMCYCLE} = \{\langle G\rangle : G \text{ tiene un ciclo hamiltoniano}\}.$$

Teorema. HAMCYCLE es NP-completo.

Definición (Traveling Salesman)**. Input:** Las distancias d entre n ciudades y una cota D.
Output: ¿Hay un tour de largo total a lo más D?
Un tour visita cada ciudad una vez y vuelve al inicio.

$$\text{TSP} = \{\langle d, D\rangle : \text{hay un tour de largo total } \leq D\}.$$

Teorema. TSP es NP-completo.

Definición (Hamiltonian Path)**. Input:** Un grafo dirigido G.
Output: ¿Hay un camino que visite cada vértice exactamente una vez?

$$\text{HAMPATH} = \{\langle G\rangle : G \text{ tiene un camino hamiltoniano}\}.$$

Teorema. HAMPATH es NP-completo.

Ejercicios

1. Para los siguientes problemas de una potencial aplicación, escribalos como lenguajes formales y demuestre su NP-completitud.

   a) Dado un grafo $G = (V, E)$ con pesos en los vértices $(w(v))_{v \in V}$, decidir si hay un cubrimiento por vértices de peso total a lo más B.

   b) Dado un conjunto U, una colección de conjuntos $A_1, \ldots, A_m \subseteq U$ y un número k, ¿existe un subconjunto $U' \subseteq U$ de tamaño a lo más k que intersecte cada A_i?

c) Dado un grafo dirigido G y un número k, ¿hay un camino que visite al menos k vértices sin repetir ninguno?

d) Dada una conjunción de cláusulas, cada una con k literales de variables distintas, ¿es satisfacible?

2. Usted trabaja en una empresa de desarollo de software y clientes le presentan los siguientes problemas para su consideración. ¿Cúal sería su recomendación profesional? Justifique su respuesta con un argumento de complejidad computacional.

   a) Una tienda guarda sus operaciones en una matriz A: sus filas son clientes y sus columnas son productos. Los elementos de la matriz indican cuántas unidades de cada producto compró cada cliente.

   La empresa está interesada en encontrar un grupo representativo de clientes: esto es, un grupo de clientes que no hayan comprado los mismos productos. En particular, la empresa quiere que ustedes diseñen un algoritmo que, dado un número k, decida si hay al menos k clientes representativos.

   b) Una empresa de telecomunicaciones administra una red dirigida compuesta por servidores y enlaces. Los usuarios de esa empresa solicitan *rutas*: cada ruta corresponde a un camino dirigido en la red, desde un servidor de origen hasta un servidor de destino. Por motivos de capacidad, la empresa no puede aceptar dos solicitudes de rutas que compartan un mismo servidor.

   La empresa quiere que usted diseñe un algoritmo que, dadas la red, las rutas solicitadas y un número k, decida si se pueden aceptar al menos k solicitudes simultáneamente.

   c) Una empresa de contenidos quiere restringir el acceso gratuito a su sitio web. Las páginas y los enlaces forman un grafo dirigido. La empresa ha definido zonas temáticas, cada una de ellas un conjunto de páginas; una misma página puede pertenecer a varias zonas. Durante una sesión, un visitante sin suscripción puede visitar a lo más una página de cada zona.

   La empresa quiere que usted diseñe un algoritmo que, dadas la web, sus zonas, una página de entrada s y una página objetivo t, decida si un visitante sin suscripción puede llegar de s a t siguiendo los enlaces y respetando las restricciones de acceso.