Práctico: Reducciones NP-Completas

Teoría de la computación

Noviembre 2025

1. Definiciones de Problemas Clásicos

Definición 1.1 (Subset Sum / Suma de Subconjuntos (SS)). *Dado un conjunto de enteros positivos $S = x_1, x_2, \ldots, x_n$ y un entero t, determinar si existe un subconjunto $S' \subseteq S$ tal que la suma de los elementos en S' sea exactamente t.*

Definición 1.2 (Suma de Aristas (SA)). *Dado un grafo $G = (V, E)$ y un entero k, determinar si existe un subconjunto de aristas cuya suma de pesos sea exactamente k.*

Definición 1.3 (Independent Set (IS)). *Dado un grafo $G = (V, E)$ y un entero k, determinar si existe un conjunto de k vértices tal que no hay ninguna arista entre ellos.*

Definición 1.4 (Clique). *Dado un grafo $G = (V, E)$ y un entero k, determinar si existe un subconjunto de k vértices tal que todo par de ellos está conectado por una arista (forman un subgrafo completo).*

Definición 1.5 (Vertex Cover (VC)). *Dado un grafo $G = (V, E)$ y un entero k, determinar si existe un conjunto de k vértices tal que toda arista del grafo tiene al menos uno de sus extremos en el conjunto.*

Definición 1.6 (Set Cover (SC)). *Dado un universo $U = e_1, \ldots, e_n$ y una colección de subconjuntos $\mathcal{S} = S_1, \ldots, S_m$, y un entero k, determinar si existe una subcolección de a lo sumo k conjuntos cuya unión cubra todo U.*

Definición 1.7 (Partition). *Dado un conjunto de enteros positivos $S = x_1, \ldots, x_n$, decidir si existe una partición de S en dos subconjuntos con suma total igual.*

Definición 1.8 (SAT). *Dada una fórmula booleana en forma normal conjuntiva (CNF), decidir si existe una asignación de valores de verdad a sus variables que haga que la fórmula sea verdadera.*

Definición 1.9 (3SAT). *SAT restringido a fórmulas en CNF donde cada cláusula tiene exactamente 3 literales.*

Definición 1.10 (3-Color (3COLOR)). *Dado un grafo no dirigido, decidir si es posible asignar uno de tres colores a cada vértice de forma que vértices adyacentes reciban colores distintos.*

Definición 1.11 (k-Color (COLOREO)). *Dado un grafo no dirigido y un entero k, decidir si es posible asignar uno de k colores a cada vértice de forma que vértices adyacentes reciban colores distintos.*

Definición 1.12 (Hamiltonian Cycle (HC)). *Dado un grafo no dirigido $G = (V, E)$, determinar si existe un ciclo que pase exactamente una vez por cada vértice y regrese al vértice inicial.*

Definición 1.13 (Knapsack (Mochila)). *Dado un conjunto de objetos con pesos $w_1, \ldots, w_n$ y valores $v_1, \ldots, v_n$, una capacidad W y un valor objetivo V, determinar si existe un subconjunto de objetos cuya suma de pesos no exceda W y cuya suma de valores sea al menos V.*

Definición 1.14 (3-Dimensional Matching (3DM)). *Dados tres conjuntos disjuntos X, Y, Z de igual tamaño y una colección $T \subseteq X \times Y \times Z$ de tripletas, determinar si existe un subconjunto $M \subseteq T$ de tamaño $|X|$ tal que no comparten elementos entre tripletas.*

Definición 1.15 (Bin Packing). *Dados n objetos con tamaños $s_1, \ldots, s_n \in (0, 1]$ y un entero k, determinar si es posible asignar los objetos a k bins de capacidad 1 sin sobrepasarla.*

2. Ejercicio Introductorio

Antes de comenzar con las reducciones más complejas, presentamos un ejemplo introductorio sencillo con su solución para ilustrar el formato esperado y los pasos típicos de una reducción.

Ejercicio 0. Demostrar que $3SAT \leq_p SAT$.

Solución.
Recordemos que:

- $3SAT$ es el problema de decidir la satisfacibilidad de fórmulas booleanas en CNF donde cada cláusula tiene exactamente 3 literales.
- SAT es el problema de decidir la satisfacibilidad de fórmulas booleanas en CNF sin restricción en el número de literales por cláusula.

Para reducir $3SAT$ a SAT, notamos que toda instancia de $3SAT$ es ya una instancia válida de SAT, ya que las fórmulas en $3SAT$ son un caso particular de las de SAT.

Entonces, la función de reducción f simplemente toma como entrada la fórmula ϕ de $3SAT$ y la devuelve sin modificaciones. Es decir:

$$f(\phi) = \phi.$$

Claramente, f se computa en tiempo constante ($O(1)$). Además, ϕ es satisfacible como fórmula de $3SAT$ si y solo si $f(\phi)$ es satisfacible como fórmula de SAT.

Por lo tanto, hemos demostrado que $3SAT \leq_p SAT$.

3. Ejercicios de Reducción

Ejercicio 1. Demostrar $SAT \leq_p 3SAT$

Ejercicio 2. Demostrar $3SAT \leq_p Clique$

Ejercicio 3. Demostrar $SAT \leq_p Clique$ (puede ser útil basarse en las partes anteriores)

Ejercicio 4. Demostrar $Clique \leq_p IS$

Ejercicio 5. Demostrar $IS \leq_p Clique$

Ejercicio 6. Demostrar $IS \leq_p VC$

Ejercicio 7. Demostrar $VC \leq_p SS$.

Ejercicio 8. Demostrar $SS \leq_p Partition$.

4. Ejercicios estilo tarea/parcial

Problema del Profesor Desorganizado (PD)

Definición. Un profesor tiene un conjunto de tareas $T = \{t_1, t_2, \ldots, t_n\}$ y un conjunto de fechas disponibles $D = \{d_1, d_2, \ldots, d_m\}$. Algunas tareas no pueden realizarse en la misma fecha debido a incompatibilidades (por ejemplo, por requerir el mismo equipamiento o recursos limitados). Estas incompatibilidades están dadas por un conjunto $I \subseteq T \times T$, donde cada par $(t_i, t_j) \in I$ indica que t_i y t_j no pueden realizarse el mismo día.

Problema: ¿Existe una asignación de tareas a fechas, $f : T \to D$, tal que para todo $(t_i, t_j) \in I$, se cumpla que $f(t_i) \neq f(t_j)$?

1. Demostrar que PD pertenece a la clase NP.
2. Demostrar que PD es NP-completo mediante una reducción desde el problema Graph Coloring.
3. ¿Qué podrías concluir si mañana se encontrara un algoritmo polinomial que resolviera PD?

Pista: Modelar las tareas como vértices de un grafo $G = (T, I)$, donde cada arista conecta tareas incompatibles. El número de fechas disponibles m representa el número de colores. La pregunta es equivalente a determinar si el grafo puede colorearse con m colores.

(Examen Setiembre 2025 Ej 4): Puzzle del Transporte de Materiales (PTM)

Definición. Se dispone de un conjunto de materiales $M = \{m_1, m_2, \ldots, m_n\}$, donde cada material m_i tiene un peso $w_i \in \mathbb{N}$. También se cuenta con un número fijo k de vehículos, cada uno con capacidad máxima $W \in \mathbb{N}$. El objetivo es asignar cada material a un vehículo, de modo que la suma de pesos transportados por cada vehículo no exceda su capacidad.

Problema: ¿Existe una función $A : M \to \{1, 2, \ldots, k\}$ tal que, para todo vehículo $j \in \{1, \ldots, k\}$, se cumpla:

$$\sum_{m_i \in A^{-1}(j)} w_i \leq W$$

y cada material sea asignado exactamente a un único vehículo?

1. Demostrar que PTM pertenece a la clase NP.
2. Demostrar que PTM es NP-completo mediante una reducción desde Bin Packing.
3. ¿Qué podrías concluir si mañana se encontrara un algoritmo polinomial que resolviera PTM?

Pista: Considerar a los materiales como objetos y a los vehículos como cajas del problema de Bin Packing. El desafío consiste en empaquetar todos los objetos en k cajas sin exceder la capacidad máxima.