Práctica 3: NP completitud

Compilado: 19 de abril de 2025

1. Determinar cuáles de las siguientes afirmaciones son verdaderas y cuáles falsas. Demostrar aquellas que son verdaderas y dar contraejemplos para aquellas que son falsas.

   a) $\mathsf{P} \subseteq \mathsf{NP}$ y $\mathsf{P} \subseteq \mathsf{coNP}$.

   b) Si $\mathsf{P} = \mathsf{NP}$, entonces $\mathsf{coNP} = \mathsf{NP}$.

   c) Si $\mathsf{P} = \mathsf{NP}$, entonces todos los lenguajes pertenecen a P.

   d) Si $\mathsf{coNP} = \mathsf{NP}$, entonces $\mathsf{SAT} \in \mathsf{coNP}$.

   e) Si $\mathsf{coNP} \subseteq \mathsf{NP}$, entonces $\mathsf{NP} = \mathsf{coNP}$.

2. ¿Es cierto que si dos lenguajes Π y Γ pertenecen a NPC entonces $\Pi \leq_p \Gamma$, y también $\Gamma \leq_p \Pi$? Justificar.

3. Sean Π y Γ dos lenguajes tales que $\Pi \leq_p \Gamma$. ¿Qué se puede inferir?

   a) Si $\Pi \in \mathsf{P}$ entonces $\Gamma \in \mathsf{P}$.

   b) Si $\Gamma \in \mathsf{P}$ entonces $\Pi \in \mathsf{P}$.

   c) Si $\Gamma \in \mathsf{NPC}$ entonces $\Pi \in \mathsf{NPC}$.

   d) Si $\Pi \in \mathsf{NPC}$ entonces $\Gamma \in \mathsf{NPC}$.

   e) Si $\Gamma \in \mathsf{NPC}$ y $\Pi \in \mathsf{NP}$ entonces $\Pi \in \mathsf{NPC}$.

   f) Si $\Pi \in \mathsf{NPC}$ y $\Gamma \in \mathsf{NP}$ entonces $\Gamma \in \mathsf{NPC}$.

   g) Π y Γ no pueden pertenecer ambos a NPC.

4. Decir si las siguientes afirmaciones son verdaderas o falsas:

   a) Si $\mathsf{P} = \mathsf{NP}$, entonces todo problema NP-completo es polinomial.

   b) Si $\mathsf{P} = \mathsf{NP}$, entonces todo problema NP-hard es polinomial.

   c) Si las clases NP-completo y coNP-completo son disjuntas entonces $\mathsf{P} \neq \mathsf{NP}$.

   d) HALTING es NP-hard y coNP-hard.

5. Suponiendo que $\mathsf{P} = \mathsf{NP}$, diseñar un algoritmo polinomial que dada una fórmula booleana ϕ encuentre una asignación que la satizfaga, si es que ϕ es satisfacible.

6. Suponiendo que $\mathsf{P} = \mathsf{NP}$, diseñar un algoritmo polinomial que dado un grafo G retorne una clique de tamaño máximo de G.

7. Sabiendo que CLIQUE es NP-completo, demostrar que SUBGRAPH ISOMORPHISM es NP-completo.

8. Considerar los siguientes dos lenguajes:

   - SHORTEST PATH (SP) $= \{\langle G, s, t, k\rangle : G$ es un grafo pesado con dos nodos s y t tales que hay un recorrido de s a t de peso menor o igual a $k\}$

- ELEMENTARY SP= $\{\langle G, s, t, k\rangle : G$ es un grafo pesado con dos nodos s y t tales que hay un camino simple de s a t de peso menor o igual a $k\}$

Demostrar que ELEMENTARY SHORTEST PATH es NP-completo y que SHORTEST PATH está en P. ¿Cuál de los dos problemas resuelven los algoritmos de camino mínimo vistos en TDA?

9. Probar que $\mathcal{L} = \emptyset$ y $\mathcal{L}^c$ no son NP-completos.

10. Considerar los siguientes dos lenguajes:

    - SUBSET-SUM $= \{\langle v_1, \ldots, v_n, k\rangle :$ existe un subconjunto $V \subseteq \{v_1, \ldots, v_n\}$ tal que $\sum_{v \in V} v = k\}$
    - UNARY-SUBSET-SUM $= \{\langle v_1, \ldots, v_n, 1^k\rangle : \langle v_1, \ldots, v_n, k\rangle \in$ SUBSET-SUM$\}$

    a) Probar que SUBSET-SUM $\in$ NPC.
    b) Probar que UNARY-SUBSET-SUM $\in$ P.
    c) Concluir que la codificación de los números afecta la complejidad de los problemas. En general, si un problema sigue siendo NP-completo cuando los números de la entrada se representan en unario entonces el problema se considera **fuertemente** NP-completo.

11. El problema DOUBLE-SAT consiste en deteminar si una formula proposicional ϕ tiene al menos dos valuaciones que la satisfacen. Demostrar que DOUBLE-SAT es NP-completo.

12. Probar que k-CLIQUE está en P para cualquier $k \in \mathbb{N}$. Concluir que dejar parámetros fijos puede cambiar la complejidad de los problemas.

13. El problema HALF-CLIQUE consiste en determinar si un grafo G de tamaño n tiene un completo de tamaño $n/2$. Sabiendo que CLIQUE es NP-completo, demostrar que HALF-CLIQUE es NP-completo. ¿Por qué este resultado no contradice el hecho de que k-CLIQUE es polinomial para todo k?

14. Demostrar que TAUTOLOGY es coNP-completo.

15. El objetivo de este ejercicio es demostrar de forma guiada que PRIME está en NP $\cap$ coNP[1].

    (*a*) Probar que UNARY PRIME $= \{1^n : n \in \mathbb{N}$ es primo$\}$ está en P.
    (*b*) Probar que PRIME $= \{n \in \mathbb{N} : n$ es primo$\}$ está en coNP.
    Para el siguiente ejercicio considerar el Teorema de Lucas: para todo número natural $n \geq 2$ vale que n es primo si y solamente si existe un a con $0 \leq a \leq n-1$ tal que $a^{n-1} \equiv 1 \mod n$ y $a^{(n-1)/q} \not\equiv 1 \mod n$ para todo divisor no trivial q de $n-1$.[2]
    (*c*) Probar que en el teorema de Lucas alcanza con pedir que $a^{(n-1)/q} \not\equiv 1$ para todo divisor q **primo** no trivial de $n-1$, en vez de para todo divisor no trivial.
    (*d*) Probar que PRIME está en NP. **Ayuda**: Observar que en el certificado podemos pedir a y la factorización en primos de $n - 1 = \prod_{i=1}^{k} p_i^{m_i}$, pero... ¿Cómo certificamos que estos números $p_1, \ldots p_k$ son a su vez primos?

[1] Se sabe que PRIME está en P desde 2002
[2] Observar que la ida de este teorema es el pequeño teorema de Fermat.