Trabajo Práctico Nro 6

Ejercicio 1. Responder breve y claramente:

a. Dar un esquema de prueba de la transitividad de las reducciones polinomiales. Ayuda: en clase hicimos lo propio con las reducciones generales.

Para probar la transitividad de las reducciones polinomiales debemos mirar al transitividad de las reducciones generales:

- Las reducciones son transitivas porque una reducción de un L1 a un L3 la reducción es la composición de dos reducciones, de L1 a L2 y L2 a L3.
  Para probar la transitividad de las polinomiales deberíamos probar que la composición de 2 reducciones polinomiales también se hace en tiempo polinomial, o sea es como decir:
- Si compongo una reducción de L1 a L2 que tarda n y otro de L2 a L3 que también tarda n, el O de las 2 combinadas es n.

b. ¿Cuándo un lenguaje es NP-difícil y cuándo es NP-completo?

Un lenguaje NP-completo cumple 2 condiciones:

- L pertenece a NP
- Todos los demás lenguajes en NP se pueden reducir a L en tiempo polinomial.
  Un lenguaje NP-difícil solo cumple lo segundo, todos los lenguajes en NP se reducen a L.
  Es decir un lenguaje NP-completo es NP difícil y esta en NP.

c. ¿Por qué si P ≠ NP, un lenguaje NP-completo no pertenece a P?

Un lenguaje NP completo pertenece NP solamente por que sino todos los lenguajes de NP se podrían resolver en tiempo polinomial sin certificado, es decir de que NP-Completo L1 pertenezca a P:

- Reduzco un Li que pertenece a NP a L1.
- Ahora puedo resolver L1 en tiempo polinomial sin certificado, ya que L1 es de P.
- Esto resolvería todos los problemas.

d. Enunciar el esquema visto en clase para agregar un lenguaje a la clase NPC.

- Debo probar que un lenguaje L1 esta en NP.
- Debo probar que hay un lenguaje NP-Completo y lo puedo reducir a L1 (probar NP-Difícil, por transitividad si puedo reducir un NP-Completo a L1 puedo reducir todos los lenguajes de NP a L1).

e. ¿Cuándo se sospecha que un lenguaje de NP está en NPI?

Se sospecha cuando:

- No estaría en P.
- No estaría en NPC.
- Esta en NP.

Ejercicio 2. Probar:

a. Si L1 ∈ NPC y L2 ∈ NPC, entonces L1 <=P L2 y L2 <=P L1.

Me esta pidiendo probar que dos lenguajes NPC completos son reducibles entre si son reducibles en tiempo polinomial de forma simétrica, es decir de L1 a L2 y viceversa.

- Imaginemos que no se puede.
- Esto implica:
  - L1 pertenece a NP y todo lenguaje Li de NP es reducible a L1.
  - L2 pertenece a NP y todo lenguaje Li de NP es reducible a L2.
  - Como L2 pertenece a NP y L1 pertenece a NP, L2 <= L1 y L2 <= L1, contradicción.
  La **definición misma de NP-completitud** garantiza que **todos los lenguajes NP-completos se reducen entre sí en tiempo polinomial**.

b. Si L1 <=P L2, L2 <=P L1, y L1 ∈ NPC, entonces L2 ∈ NPC. Ayuda: recurrir directamente a la definición de NPC.

Para que L2 este en NPC debe pertenecer 1 NP y que se puedan reducir todos los lenguajes de NP a NPC:

- L2 esta en NP porque L1 esta en NP y al reducir en tiempo polinomial L2 a L1 lo podes resolver en NP.
- L2 esta en NPC, porque al reducir L1 a L2 en tiempo polinomial estamos diciendo implícitamente que puedo reducir cualquier lenguaje en NP a L2 en tiempo polinomial por transitividad, puesto que todos los lenguajes de NP se pueden reducir a L1.

Ejercicio 3. Un lenguaje es CO-NP-completo sii todos los lenguajes de CO-NP se reducen polinomialmente a él. Probar que SATC es CO-NP-completo. Ayuda: L1 <= L2 sii L1 C <= L2 C.

Siguiendo la propiedad de las reducciones, donde si puedo reducir un lenguaje a otro, el complemento del lenguaje es reducible al complemento del otro:

- SAT ∈ NP → su complemento (SATC) ∈ CO-NP.
- SATC es CO-NP-difícil

- Todos los lenguajes de NP son reducibles a SAT en tiempo polinomial, puesto que esta en NP.
- Todos los lenguajes de CO-NP son reducibles a SATC por propiedad.
- Esto prueba que SATC es CO-NP completo.

Ejercicio 4. Sean los lenguajes A y B, tales que A ≠ □, A ≠ Σ*, y B ∈ P. Probar: (A ∩ B) ≤P A. Ayuda: intentar con una reducción polinomial que, dada una cadena w, lo primero que haga sea chequear si w ∈ B, teniendo en cuenta que existe un elemento e que no está en A.

La reducción es simple:

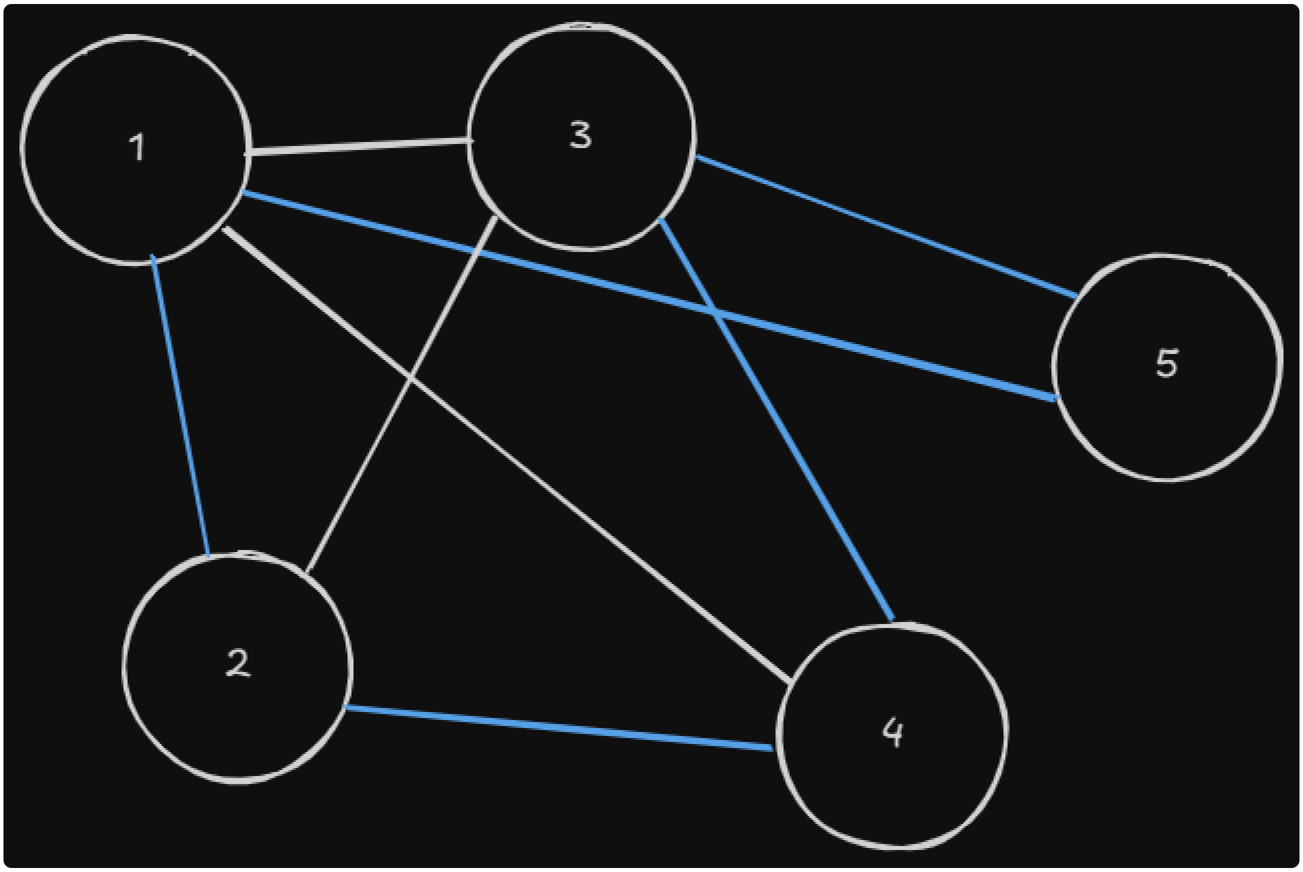
- Cadena que esta en B (ejecutamos MT B sobre la misma y termina en tiempo polinomial) retorna la cadena así como esta.
- No esta en B, retornamos algo fuera de A.
  Lo de estar en B funciona porque aplicar una MT B sobre la misma tarda tiempo polinomial, esto nos confirma que puede estar en la intersección, al pasárselo luego a la MT que decide A determinara si además esta en A y por tanto:
- Si esta en A también en la intersección, y por tanto es correcto.
- Sino estaba en A entonces se rechaza, correcto.
  Además es importante notar porque si no esta en B retornamos algo fuera de A:
- Al no estar en B y retornar algo fuera de A cumplimos lo de adentro adentro y lo de afuera afuera.
- Si algo esta en A y no en B, no entra en la intersección.

Ejercicio 5. Sea el lenguaje SH-s-t = {(G, s, t) | G es un grafo no dirigido y tiene un camino de Hamilton del vértice s al vértice t}. Un grafo G = (V, E) tiene un camino de Hamilton del vértice s al vértice t sii G tiene un camino entre s y t que recorre todos los vértices restantes una sola vez. Probar que SH-s-t es NP-completo. Ayuda: se sabe que CH, el lenguaje correspondiente al problema del circuito hamiltoniano, es NP-completo.

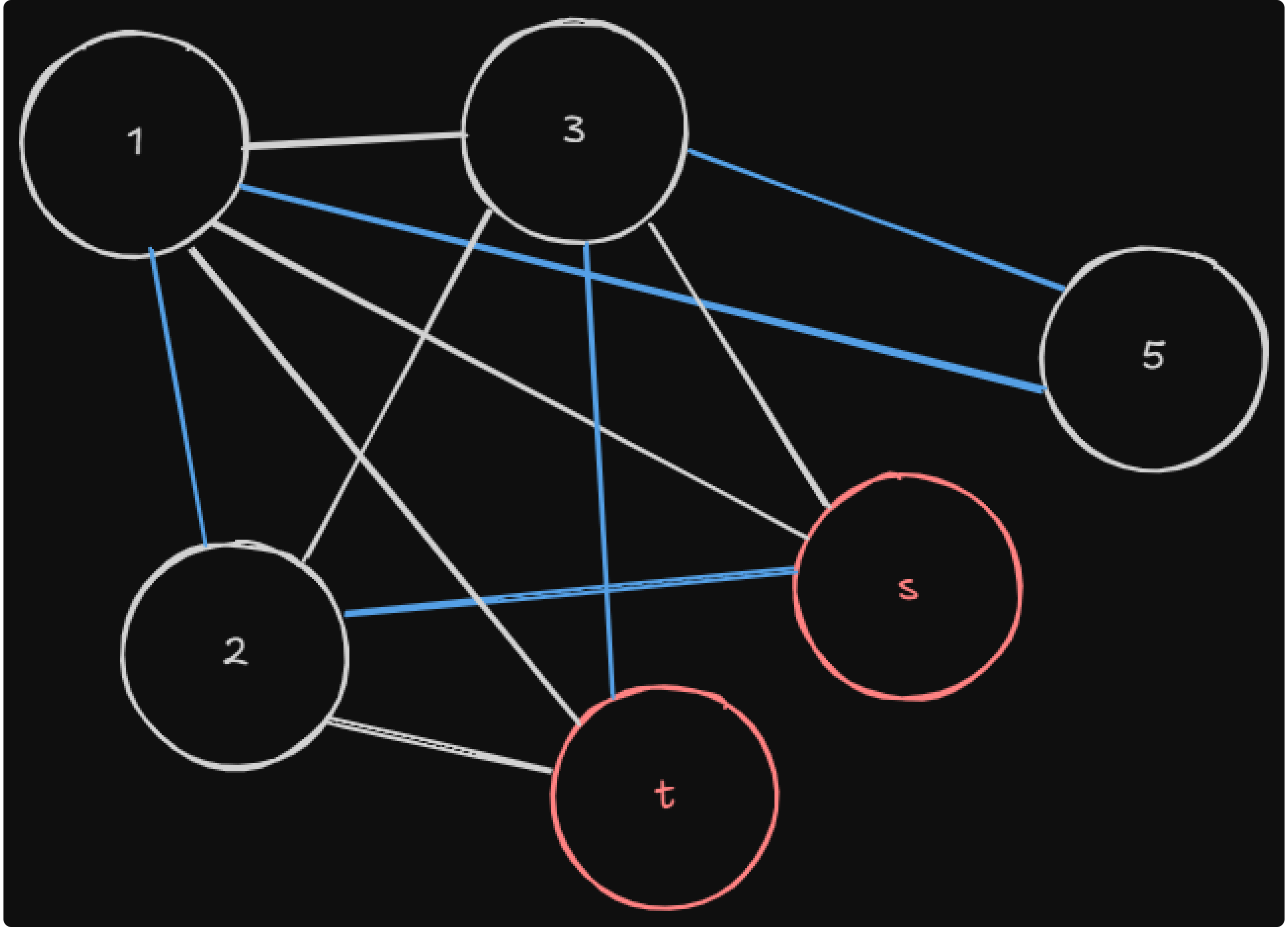
Por lo que dijeron los ayudantes:

- No basta con seleccionar cualquier nodo del grafo de CH para el camino, ya que si el principio y el final es el mismo se cuenta como que paso por el mismo nodo 2 veces.
- Además dijo que habría que sacar o quitar algo.
  En mi opinión, la reducción puede ser lo siguiente:
- Seleccionamos un nodo cualquiera v0 y lo borramos.

- A los vecinos de v0 le conectamos s y t.
  Esto lo soluciona porque el circuito es tal que así:
  v0 -> vecino -> ... -> vecino -> v0
  (Suponiendo que arranquemos en v0, que podría ser cualquiera)
  Entonces al conectar s a todos los vecinos, puedo continuar la ruta, y al llegar al vecino anteúltimo del circuito en cambio va a t.
  Ejemplo:



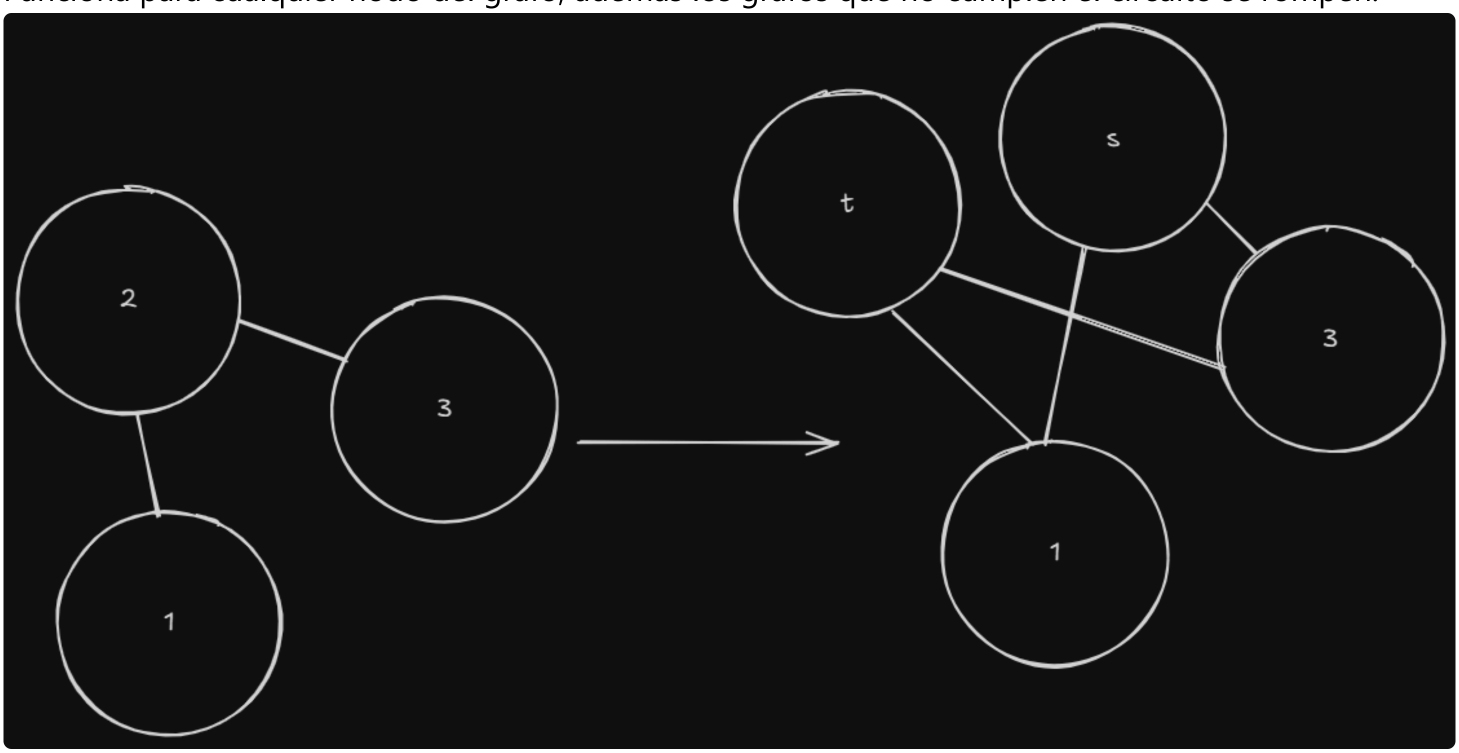
  Este ejemplo de la teoría, imagínate la ruta 4 (v0)->2 (vecino) -> 1->5->3 (vecino) -> 4.
  Reducido:



Camino de Hamilton:

s->2->1->5->3->t

Funciona para cualquier nodo del grafo, además los grafos que no cumplen el circuito se rompen:



Como vemos podría parecer que funciona pero, dado que hay que arrancar desde s, no hay ruta que termine en t sin pasar por un nodo devuelta.

Técnicamente, al ser un grafo ciclo tiene un camino de hamilton, pero no uno que arranque en s y termine en t. Si arrancamos en s, terminamos si o si en 1 y 3.

Esto prueba que el camino de hamilton es NP-difícil, además, es de NP porque:

- El certificado es sucinto, seria un camino, tamaño n.
- Hay que recorrerlo y verificar condiciones, esto es polinomial.