# PRÁCTICA 6

Complejidad temporal.

**Ejercicio 1. Asumamos que Ʃ \* solo tiene cadenas de unos y ceros, pudiendo incluir la cadena vacía λ. En este contexto, dada una cadena w de Ʃ \*, se define como E (w) a su cadena espejo, que es la que se obtiene reemplazando en w los unos por ceros y los ceros por unos (p.ej. E (1001 ) = 0110 y E (λ) = λ), y también se define que L es un lenguaje espejo si se cumple, para toda cadena con distintas de λ, que w ∈ L↔E (w) ∈ Lᶜ. Sea f la función que transforma todo w en E (w). Respondedor, justificando las respuestas:**

1. **¿Es f una función total computable?**

Si, dado que existe una MT que computa tal función en su cinta de salida y se detiene. Un ejemplo de tal máquina podría ser una que si detecta que el caracter es 0 lo transforma en 1; caso contrario, transforme en 0. Luego, que avance a la derecha. Si encuentra un B, significa que ha concluido el procesamiento de w. Eventualmente, por más grande que sea w, f terminará de computarlo, por lo que la función es, a priori, total computable.

1. **¿Cuánto tarda una MT que computa f?**

Dado que lo único que hace es invertir, tarda n (es lineal).

1. **Si L es un lenguaje espejo ¿Se cumple que f es una reducción polinomial de L a Lᶜ?**

No, la cadena vacía genera problemas. El lenguaje E (w) puede definirse como su cadena espejo, excepto para λ. El problema va a ser que, cuando el input sea λ, el resultado va a ser λ. El problema es que E (λ) = λ, no puede estar en los dos conjuntos.

## Ejercicio 2. Hemos construido en la Clase 6 una reducción polinomial del problema CH (circuito de Hamilton) al problema TSP (viajante de comercio), es decir CH αP TSP.

### Sabiendo que TSP es NP-completo y sin asumir nada sobre CH:

* 1. **¿Se cumple que CH ∈ NP?** Si, porque CH es tan o menos complejo que TSP, y TSP está en NP, por lo tanto CH está en P o NP.
  2. **¿Se cumple que CH ∈ NPC?** No, lo único que se de CH es ∈ NP, no se si todo lenguaje de NP se reduce a él.

1. **Sabiendo que CH es NP-completo y sin asumir nada sobre TSP:**
   1. **¿Se cumple que TSP ∈ NP?** No necesariamente, TSP podría estar en RE por ejemplo podría ser HP.
   2. **¿Se cumple que TSP ∈ NPC?** Idem anterior, además no puedo confirmar que TSP ∈ NP

**a) Sabiendo que TSP es NP-completo y sin asumir nada sobre CH:**

**i. ¿Se cumple que CH ∈ NP?**

Por definición de NP-Completo, un lenguaje L que se encuentra dentro de  esta clase tiene que cumplir como condición:

1. Pertenecer a NP
2. Para todo L’ ∈ NP, L’ pueda reducirse en tiempo eficiente (poly(n)) a L.

Además, por propiedad de la reducción comprobamos que si reducimos L1 a L2, y L2 ∈ NP → L1 ∈ NP.

Por lo cual, CH si pertenece a la clase NP.

**ii. ¿Se cumple que CH ∈ NPC?**

No podemos asegurarlo con los datos provistos por el enunciado. Ya sabemos que CH efectivamente pertenece a la clase NP. Queda probar que todo lenguaje L puede reducirse en tiempo eficiente a CH. Si pudiésemos reducir TSP αP CH correctamente, por ejemplo, entonces quedaría comprobado que si se cumple.

**b) Sabiendo que CH es NP-completo y sin asumir nada sobre TSP:**

**¿Se cumple que TSP ∈ NP?  ¿Se cumple que TSP ∈ NPC?**

Dada la reducción CH αP TSP, sabemos que TSP tiene que ser “tan o más complejo” que CH. Además por enunciado sabemos que CH es NP-Completo.

Con el teorema: Sean L1 ∈ NPC y L2 ∈ NP, tales que L1 αP L2, entonces L2 ∈ NPC.

A partir de esto podemos deducir que TSP pertenece a NPC y por consiguiente también pertenece a NP.

## Ejercicio 3. En la Clase 6 quedó planteado como ejercicio probar que si L1 αP L2, L1 ∈ NPC y L2 ∈ NP, entonces L2 ∈ NPC. Probarlo.

1. L1 ∈ NPC (hipotesis)
2. L1 ∈ NP (por 1)
3. V L’ ∈ NP , L’ αP L1 (por 1)
4. L2 ∈ NP (hipotesis)
5. L1 αP L2 (hipotesis)
6. V L’ ∈ NP , L’ αP L2 (por transitividad de reducciones, 5)
7. L2 ∈ NPC (por 4 y 6)

*Sabemos que L1 pertenece a NPC por enunciado. Por ende, se sabe que también pertenece a NP. Sabemos que L2 pertenece a NP por enunciado.*

*El lenguaje L1 se reduce polinomialmente a L2, por ende, L2 es igual o más complejo que L1. Es decir que si L2 pertenece a NP, como sabemos que es igual o más complejo que L1, y L1 pertenece a NPC, L2 pertenece a NPC.*

A partir de la imagen de arriba se logra SAT αP L. Cómo SAT ∈ NPC, cualquier lenguaje L’ ∈ NP puede reducirse en tiempo eficiente a SAT.

Entonces por propiedad de transitividad se comprueba que si: L’ αP SAT y SAT αP L, entonces L’ αP L.

Esto mismo puede aplicarse para L1 y L2 del enunciado.

L2 entonces cumple con la definición de NPC. Pertenece a NP y cualquier L perteneciente a NP puede reducirse a él.

## Ejercicio 4. En la Clase 6 quedó planteado como ejercicio probar que si L1 ∈ NPC y L2 ∈ NPC, entonces se cumple tanto L1 αP L2 como L2 αP L1. Probarlo.

Los problemas NPC son los más difíciles desde el punto de vista de la complejidad temporal (dentro de la clase NP). Como mencionamos antes, en su definición se especifica que dado un lenguaje L ∈ NPC entonces:

1. L debe pertenecer a NP
2. Para todo L’ ∈ NP, L’ pueda reducirse en tiempo eficiente (poly(n)) a L.

Dada esta definición si L1 ∈ NPC y L2 ∈ NPC

L1 αP L2: Se debe cumplir ya que L1 pertenece a la clase NP por la parte 1 de la definición y L2 tiene que cumplir que cualquier L ∈ NP pueda reducirse en tiempo eficiente sobre él.

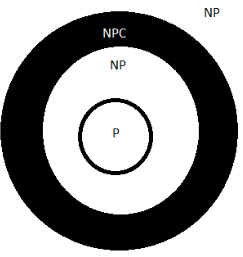
De manera similar, lo mismo debe complirse para L2 αP L1.

L1 ∈ NPC → L1 ∈ NP → L2 ∈ NPC→ L2 -∈ NP → L1 αP L2 → L2 αP L1

## L2 ∈ NPC → L2 ∈ NP → L1 αP L2 → L1 ∈ NPC → L1 ∈ NP → L2 αP L1

L1 ∈ NPC → L1 ∈ NP → L1 αP L2 → L2 ∈ NPC → L2 ∈ NP → L2 αP L1

1. *L1 ∈ NPC (hipótesis)*
2. *L2 ∈ NPC (hipótesis)*
3. *∀ L’ ∈ NP , L’ αP L1 (por 1)*
4. *L1 ∈ NP (por 1)*
5. *∀ L’ ∈ NP , L’ αP L2 (por 2)*
6. *L2 ∈ NP (por 2)*
7. *L1 αP L2 (por 5 y 4)*
8. *L2 αP L1 (por 3 y 6)*



Sabemos que L1 pertenece a NPC por enunciado. Por ende, sabemos que todo lenguaje perteneciente a NP, puede reducirse polinomialmente a L1. Además, como L1 pertenece aNPC, sabemos que pertenece a NP.

Sabemos que L2 pertenece a NPC por enunciado. Por ende, sabemos que todo lenguaje perteneciente a NP, puede reducirse polinomialmente a L2. Además, como L2 pertenece a NPC, sabemos que pertenece a NP.

L1 es tan o más complejo que L2, y a la viceversa. (¿Serían igual de complejos?)

Por lo tanto, podemos concluir que como L1 es un lenguaje perteneciente a NP, puede

reducirse polinomialmente a L2, ya que este último es completo. Lo mismo sucede al revés, L2 puede reducirse polinomialmente a L1, porque L2 pertenece a NP (explicado arriba) y L1 es NPC.

Ejercicio 5. En la Clase 6 quedó planteado como ejercicio probar la parte (b) de un teorema  
que establecía: si L1 αP L2 y L2 ∈ NP, entonces L2 ∈ NP. Probarla. Ayuda: Basarse en la parte (a)del teorema, probada en clase.Ejercicio 6. Probar que la relación αP es transitiva.  
Ejercicio 7. Probar que si se cumple L1 αP L2, L2 αP L1, y L1 ∈ NPC, entonces L2 ∈ NPC.  
Ejercicio 8. Sean los lenguajes A y B, tales que A ≠ ∅, A ≠ Ʃ\*, y B ∈ P. Probar: (A ⋂ B) αP A.  
Ejercicio 9. Sea el lenguaje HPA-s-t = {(G, s, t) | G es un grafo que tiene un camino de  
Hamilton del vértice s al vértice t}. Un grafo G = (V, E) tiene un camino de Hamilton si existe un  
par de vértices v1 y v2 de V tal que G tiene un camino entre v1 y v2 que recorre todos los  
vértices restantes una sola vez. Probar que HPA-s-t es NP-completo. Ayuda: se sabe que CH(el problema del circuito de Hamilton) es NP-completo.