

Clase 5-tiempo y espacio

martes, 8 de abril de 2025 19:02

Tiempo \rightarrow cantidad de pasos que ejecuta una MT

Espacio \rightarrow cantidad de celdas que ocupa una MT

En la práctica no me sirve que sea decidible I tarda muchísimo tiempo

Eficiente es MT con tiempo polinomial $\rightarrow n$ elevado a la k

Complejidad temporal:

Espacio hoy es casi infinito, puedo agregar mas memoria o lo que sea. Tiempo es finito, no lo puedo reutilizar y demás

MT tarda más a medida que la entrada w tardaás. Entonces el tiempo de M no se mide en términos absolutos sino en $T(n)$. Se definen en términos del tamaño $w(n=|w|)$

Una función $T_1(n)$ es del orden de una función $T_2(n)$, que se anota así: $T_1(n) = O(T_2(n))$, si para todo $n \geq n_0$ se cumple $T_1(n) \leq c \cdot T_2(n)$, con $c > 0$.

Por ejemplo (principal):

Time $T(n)$ si hay una mt M que lo decide en tiempo $O(T(n))$

Se considera el tiempo máximo, procesar cualquier cadena consume a lo sumo $O(|w|)$ pasos

Una cadena que nos cague nos caga todo

dificil sacar promedio y mínimo, pq es difícil saber qué pasa con todas las cadenas, entonces uso el máximo

Clase P : clase de lenguajes que se resuelven en tiempo polinomial

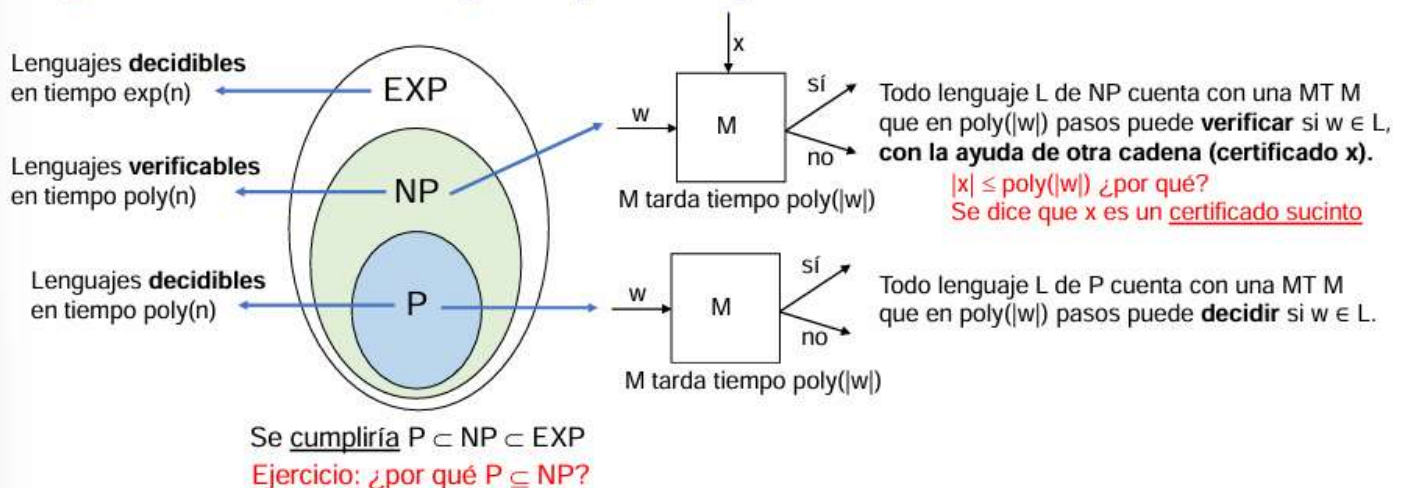
P es aceptado por $poly(n)$

Tesis fuerte de Turing: si un modelo computacional realiza algo en tiempo poli entonces hay una mt que también lo hace en ese tiempo. Hoy es mentira, pq las m cuánticas desafían la tesis.

Un nro en binario ocupa \log base 2 de n

Lenguajes que no estarían en P (se sospecha pero no se puede demostrar formalmente)

A sat y ch, si le dan un a que es una cosa a verificar, puedo chequear si es en tiempo polinomial. Lo que no hacen en polinomial es tener que chequear todas.



- En otras palabras:

Si un problema está en P , podemos asegurar que sus soluciones **se encuentran eficientemente**.

Si un problema está en NP , sólo podemos asegurar que sus soluciones **se verifican eficientemente**.

- Intuitivamente, $P \neq NP$ (encontrar una solución parece ser más difícil que verificarla). Sin embargo, al día de hoy esta relación **no ha podido ser probada**.

La definición de la derecha arriba: la máquina que verifica tarda tiempo polinomial. Si lee algo que mide más que w , se va de polinomial

Métricas de complejidad:

Dinámicas:

- consumo de recursos abstractos
- Si cambiamos muchas veces de lado el cabezal

Estáticas

- Cero fero

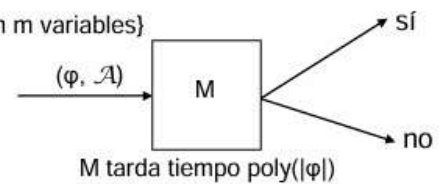
Definición de no menos intuitiva. Un probador y un verificador. En lugar de que el probado haga cosas y luego el verificador verifique va haciendo el verificado y el probador un ping pong.

- $SAT = \{\varphi \mid \varphi \text{ es una fórmula booleana satisfactible sin cuantificadores con } m \text{ variables}\}$

SAT no estaría en P: hay 2^m asignaciones \mathcal{A} para chequear.

SAT está en NP: dadas una fórmula booleana φ y una asignación \mathcal{A} , se puede verificar en tiempo $\text{poly}(n)$ si \mathcal{A} satisface φ .

¿SAT^c está en NP? ¿Cuánto mide un certificado en este caso?



Satc no estaría en NP pq el certificado mide xponencial pq para saber que no sea tenes que chequear todas las opciones

Releer la última diapo del anexo. Hay que tener cuidado cuando haband de n nro n pq lo que importa es el tamaño de la entrada