TEMA 6. Complejidad espacial.

1- Definiciones Básicas.

Utilizaremos MT con una cinta de entrada de solo lectura, sobre la que el cabezal sólo se mueve a lo largo de la cadena de entrada (más los dos blancos que la delimitan). MTD con varias cintas. El espacio consumido es el mismo utilizando MT con cualquier cantidad finita de cintas → no se pierde generalidad considerante MT con una sola cinta.

Una MT M con una cinta de entrada de sólo lectura y varias cintas de maniobra, trabaja en espacio **S(n)** ↔ para toda entrada w, |w|=n, M utiliza a lo sumo S(n) celdas en toda cinta de trabajo, en su única computación si es determinística o en cada una de sus computaciones si es no determinística.

Una MT M con una cinta de entrada de sólo lectura y varias cintas de maniobra, trabaja en espacio **O(S(n))** ↔ para toda entrada w, |w|=n, M utiliza a lo sumo O(S(n)) celdas en toda cinta de trabajo, en su única computación si es determinística o en cada una de sus computaciones si es no determinística.

Un lenguaje (problema) está en la clase **DSPACE(S(n))** ↔ existe una MTD con una cinta de entrada de sólo lectura y varias cintas de trabajo que lo reconoce (resuelve) en espacio 0(S(n)).

Un lenguaje (problema) está en la clase **NSPACE(S(n))** ↔ existe una MTND con una cinta de entrada de sólo lectura y varias cintas de trabajo que lo reconoce (resuelve) en espacio 0(S(n)).

El uso de una cinta de entrada de sólo lectura permite trabajar con orden espacial menor que lineal.

El espacio se consume en mucha menor cantidad que el tiempo, ya que el espacio puede reutilizarse y el tiempo no.

Tanto la jerarquía temporal como la espacial están incluidas en la clase R (la MT siempre para). En cualquier caso, si existe una MT que trabaja en espacio S(n)→ existe una MT equivalente que trabaja en el mismo espacio y siempre para (en un espacio acotado sólo podemos tener un número finito de configuraciones distintas de una MT a partir de una entrada).

Consideramos lo no polinomial como exponencial y tendremos la clase **EXPSPACE**.

Ejemplos:

- Problema reconocedor de palíndromos (con un separador):

Trabaja en espacio O(log n). Idea: dos cintas, en cada una, una parte. Esta técnica permite representar las posiciones de los cabezales mediante contadores. DSPACE(log n) = **DLOGSPACE**.

- Alcanzabilidad de un grafo orientado (G,v1,v2):

G es un grafo orientado, busco un camino del vértice v1 al vértice v2.
Está en NSPACE(log n). NSPACE(log n) = **NLOGSPACE**.

DLOGSPACE ⊆ NLOGSPACE

Más clases en la jerarquía espacial:

PSPACE: se resuelven en un espacio determinístico polinomial.

NPSPACE: se resuelven en un espacio no determinístico polinomial.

PSPACE = $\bigcup i \geq 0$ DSPACE(n^i)
NPSPACE = $\bigcup i \geq 0$ NSPACE(n^i)

Las funciones S(n) de buen comportamiento serán espacio-construibles.

Características de la jerarquía espacial:

- Teorema de compresión lineal: el salto de una clase espacial a otra que la incluya estrictamente también se produce por medio de una función mayor que lo determinado por factores constantes.

- Para que una clase DSPACE(S2(n)) incluya estrictamente a una clase DSPACE(S1(n)), además de S1(n) = O(S2(n)) debe darse que S2(n) sea significativamente mayor que S1(n) cuando n tiende a infinito.
  En particular DLOGSPACE ⊂ DSPACE(n^k) → DLOGSPACE ⊂ PSPACE.
- Es densa: siempre hay un lenguaje recursivo fuera de DSPACE(S(n)).
- Existe una función total computable S(n) tal que DSPACE(S(n)) = PSPACE → PSPACE ⊂ EXPSACE.

2- Relación entre jerarquía temporal y espacial.

- Si una MT tarda tiempo T(n), no recorre más de T(n) celdas.
- En espacio S(n), una MT no puede ejecutar más de $c^{S(n)}$ pasos sin repetir alguna configuración. → los problemas de DLOGSPACE son tratables pues de resuelven en tiempo determinístico polinomial. DLOGSPACE ⊆ P.

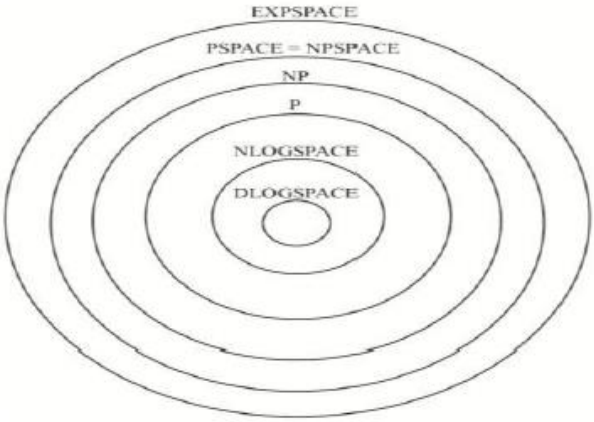
Los problemas de DLOGSPACE son problemas de P con resoluciones que consisten en validaciones de propiedades sobre componentes de sus instancias.

Teorema de Savitch: Si L ∈ NSPACE(S(n)), con S(n) una función espacio construible con S(n) ≥ log n → L ∈ DSPACE(S^2(n)). (Pasar del no determinismo al determinismo (espacial) impacta en orden cuadrático).

Por este teorema tenemos que NSPACE(n^k) ⊆ DSPACE(n^{2k}) para $k \geq 1$ → NPSPACE ⊆ PSPACE. Por definición sabemos que PSPACE ⊆ NPSPACE. Luego NPSPACE = PSPACE. Es decir, la complejidad espacial de la clase de los problemas con resolución no determinística polinomial coincide con la clase de los problemas con resolución determinística polinomial.

NP ⊆ NPSPACE (toda computación con un número polinomial de pasos no ocupa más de un número polinomial de celdas)→NP ⊆ PSPACE.

Veamos una sólo jerarquía de las distintas clases de complejidad temporal y espacial:



Probar cuáles de estas inclusiones son estrictas es un problema que está abierto. Como DLOGSPACE C PSPACE, entonces al menos una tiene que serlo.

Al contrario que en la jerarquía temporal, toda clase espacial no determinística es cerrada en la operación de complemento.

Teorema de Immermann: para toda función espacio-construible $S(n) \geq \log n$ se cumple que NSPACE(S(n)) = CO-NSPACE(S(n))

Los teoremas de Savitch e Immerman revelan que el impacto del no determinismo es mayor en la complejidad temporal que en la espacial.

3- Completitud en la jerarquía espacio-temporal.

La NP-Completitud es un caso particular del concepto de completitud en cualquier clase de problemas, temporal o espacial. Para NP, probar que un problema es NPC con respecto a las reducciones polinomiales (en tiempo) significa que no se resuelve eficientemente. El mismo criterio podemos usarlo en cualquier clase de complejidad haciendo referencia un tipo determinado de reducción.

Usaremos las **reducciones logarítmicas** (en espacio): es una m-reducción entre los lenguajes (problemas) computable en espacio determinístico logarítmico.

Reducciones logarítmicas:

Ni la cinta de entrada ni la de salida intervienen en el cálculo del espacio.

L1 αlog L2 quiere decir que existe una reducción logarítmica de L1 a L2.

Poly-time: reducciones polinómicas.

Log-space: reducciones logarítmicas.

Toda reducción log-space es una reducción poly-time pues el número de celdas de orden logaritmo requieren a lo sumo un tiempo polinomial para recorrerlas.

αlog es reflexiva y transitiva.

Las clases DLOGSPACE, NLOGSPACE, P, NP y PSPACE son cerradas con respecto a las reducciones log-space.

DEF:

Un lenguaje (problema) L es $\mathcal{C}$**-difícil** con respecto a las reducciones poly-time (respectivamente log-space) ↔ si para todo lenguaje (problema) L' de la clase $\mathcal{C}$ se cumple que L' αP L (respectivamente L' αlog L). Si L además está en la clase $\mathcal{C}$, entonces es $\mathcal{C}$**-completo.**

Ante la sospecha de que dos clases cumplen que $\mathcal{C}1$ C $\mathcal{C}2$, entonces probar que un problema es $\mathcal{C}2$-completo significa que no pertenece a $\mathcal{C}1$, a menos que $\mathcal{C}1 = \mathcal{C}2$.

La completitud se puede utilizar para probar la igualdad de dos clases: dadas dos clases $\mathcal{C}1$ y $\mathcal{C}2$, si L es $\mathcal{C}1$-completo y $\mathcal{C}2$-completo con respecto a las reducciones poly-time (respectivamente log-space), y $\mathcal{C}1$ y $\mathcal{C}2$ son cerradas con respecto a las reducciones polytime (respectivamente log-space), entonces $\mathcal{C}1 = \mathcal{C}2$.

4- Ejemplos clásicos de problemas C-completos en la jerarquía espacio-temporal.

El problema de la alcanzabilidad en grafos orientados es NLOGSPACE-completo.

Circuitos Boolenanos es P-completo log-space.

La Programación Lineal es P-completo con respecto a log-space.

El problema QBF (por *quantified boolean formulas*) es PSPACE-completo con respecto a las reducciones *poly-time* ={ determinar si una fórmula booleana con cuantificadores y sin variables libles es verdadera}

Juegos bipersonales son PSPACE-completos con respecto a poly-time.