Tema 8. Complejidad Computacional.

Tema 8. Complejidad Computacional

Índice:

- Autómatas, Complejidad y Computabilidad.
- Clasificación de los Problemas de Decisión.

Autómatas, Complejidad y Computabilidad

- En la Teoría de la Computación, los tres siguientes áreas:
  - Autómata,
  - Complejidad y
  - Computación

  están relacionados por la siguiente pregunta:
  - *¿Cuáles son las capacidades y limitaciones de los ordenadores?*
- Sin embargo, esta pregunta se interpreta de forma diferente en cada una de las 3 áreas.

Autómatas, Complejidad y Computabilidad

Teoría de Autómatas:

- Se encarga de las definiciones y propiedades de los modelos matemáticos de computación (esenciales en áreas aplicadas de la informática).
- Uno de estos modelos son los Autómatas Finitos, utilizados en:
  - Procesamiento de textos
  - Compiladores
  - Diseño Hardware.
- Otro modelo son las Gramáticas Libres de Contexto, usadas en:
  - Lenguajes de programación
  - Inteligencia Artificial.

Autómatas, Complejidad y Computabilidad

Teoría de la Complejidad:

- Se basa en tratar de dar respuesta a la siguiente pregunta:
  - *¿Qué hace a algunos problemas computacionalmente difíciles y a otros sencillos?*
- Tiene como finalidad la creación de mecanismos y herramientas capaces de describir y analizar la complejidad de un algoritmo y la complejidad intrínseca de un problema.

Autómatas, Complejidad y Computabilidad

Teoría de la Computabilidad:

- Está muy relacionado con la teoría de la Complejidad, ya que introduce varios de los conceptos que esta área utiliza.
- Su finalidad principal es la clasificación de diferentes problemas, así como formalizar el concepto de *computar.*
- Así, estudia qué lenguajes son *decidibles* con diferentes tipos de *"máquinas"* y diferentes modelos formales de computación.

Complejidad Computacional

- Estudia el orden de complejidad de un algoritmo que resuelve un problema *decidible.*
- Para ello, considera los 2 tipos de recursos requeridos durante el cómputo para resolver un problema:
  - Tiempo: Número de pasos base de ejecución de un algoritmo para resolver un problema.
  - Espacio: Cantidad de memoria utilizada para resolver un problema.

Complejidad Computacional

- La complejidad de un algoritmo se expresa como función del tamaño de la entrada del problema, n.
- Se refiere al ratio de crecimiento de los recursos con respecto a n:
  - Ratio del Tiempo de ejecución (Temporal): $T(n)$.
  - Ratio del espacio de almacenamiento necesario (Espacial): $S(n)$.

Clasificación de Problemas de decisión

- En base a dos criterios:
  - *Teoría de la Computabilidad*
    - *Decidible.*
    - *Parcialmente Decidible (reconocible).*
    - *No Decidible.*
  - *Teoría de la Complejidad Computacional*
    - *Conjuntos de Clase de Complejidad (Clase L, NL, P, P-Completo, NP, NP-Completo, NP-Duro…).*

Un problema de decisión es aquel en el que en el que las respuestas posibles son Si o No

Clasificación de Problemas de decisión

Considerando la Teoría de la Computabilidad un problema de decisión podrá ser:

- Decidible (o resoluble algorítmicamente):
  - Si existe un procedimiento mecánico (MT) que lo resuelva.
  - Además, la MT debe detenerse para cualquier entrada.
- Parcialmente Decidible (Reconocible):
  - Si existe un procedimiento mecánico (MT) que lo resuelva.
  - Además, la MT debe detenerse para aquellas entradas que son una solución correcta al problema.
- No Decidible
  - Si NO es decidible

Clasificación de Problemas de decisión

Considerando la Teoría de la Complejidad Computacional un problema de decisión podrá ser:

- *Conjuntos de Clase de Complejidad (Clase L, NL, P*, P-Completo*, NP, NP-Completo, NP-Duro…).*

En este caso, nos centraremos únicamente en los problemas denominados Clase P, NP y NP-Completo.

Sin embargo, para esta distinción es necesario considerar el modelo teórico de las **Máquinas de Turing**.

Además, debemos distinguir entre:

- **MT Determinista** (Para cada par *(estado, símbolo),* existe como máximo una transición a otro estado).
- **MT No Determinista** (Existe al menos un par *(estado, símbolo)*, con más de una transición a estados diferentes).

Clasificación de Problemas de decisión

Considerando la Teoría de la Complejidad Computacional un problema de decisión podrá ser:

- Clase P (Polynomial-time)
  - Contiene aquellos problemas de decisión que una **MT Determinista** puede resolver en **tiempo polinómico**.
    - Los problemas de complejidad polinómica son tratables, es decir en la práctica se pueden resolverse en tiempo *razonable.*
    - La mayoría de los problemas *corrientes* (ordenación, búsqueda…) pertenecen a esta clase.

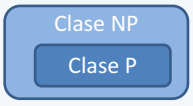
Clasificación de Problemas de decisión

Considerando la Teoría de la Complejidad Computacional un problema de decisión podrá ser:

- Clase NP (Non-Deterministic Polynomial-time)
  - Contiene aquellos problemas de decisión que una **MT No Determinista** puede resolver en **tiempo polinómico**.

Como toda MTD es un caso particular de una MTND:

$P \subseteq NP$



Saber si P=NP o P≠NP es todavía un problema abierto en computación teórica!!
Tan importante es demostrar que estas clases son distintas, que es uno de los *problemas* premiados con 1.000.000 $.

http://www.claymath.org/millennium/P_vs_NP/

Clasificación de Problemas de decisión

Considerando la Teoría de la Complejidad Computacional un problema de decisión podrá ser:

- Clase NP-Completo
  - Un problemas de decisión es NP-Completo sii:
    - Es NP
    - Todos los demás problemas de NP se pueden se pueden reducir a él en tiempo polinómico.

Reducir de un problema:
Es una manera de convertir un problema en otro de tal forma que la solución al segundo problema se puede utilizar para resolver el primero.

Tema 8. Complejidad Computacional.