.UBAfiuba
FACULTAD DE INGENIERÍA

Complejidad Algorítmica – con máquinas de Turing

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Complejidad temporal

Sea

M una TM deterministica que finaliza con todos los inputs

Llamaremos

Tiempo de ejecución o complejidad temporal de M

A la función f:N → N

Donde f(n) es el máximo numero de pasos que M usa en cualquier input de tamaño n

Diremos:

M se ejecuta en tiempo f(n)

M es una f(n)-Maquina de Turing

Notación BIG-O

El tiempo exacto de ejecución de una TM

Es difícil de calcular

Por ese motivo

se realiza una aproximación de los tiempos de ejecución

Excluyendo coeficientes de f(n)

Y quedándonos con el término de f(n) de mayor orden.

Llamamos a esa representación

Notación BIG-O

Notación BIG-O (cont.)

Sean

f y g funciones , f,g:N-→ R+

Diremos

que f(n)=O(g(n))

Si

Existe un valor c positivo y un n_0 tal que para todo $n \geq n_0$, $f(n) \leq c\ g(n)$.

Clase de complejidad temporal

Sea

t: N → R+ una función

Definimos

Una clase de complejidad temporal TIME(t(n))

A la colección

De todos los lenguajes que son decidibles por una maquina de O(t(n))-tiempo Maquina de Turing

Modelos polinomialmente equivalentes

Existen

Diversos modelos determínisticos de computo

Todos ellos

Tienen igual (o menos) poder de computo que un TM

Se puede demostrar que

Todos los modelos razonables deterministicos son polinoalmente equivalentes

Es decir que cada modelo puede similar al otro

Con solo un imcremento polinomial del tiempo de ejecución

Tiempo de ejecución en TM no determinístico

Sea

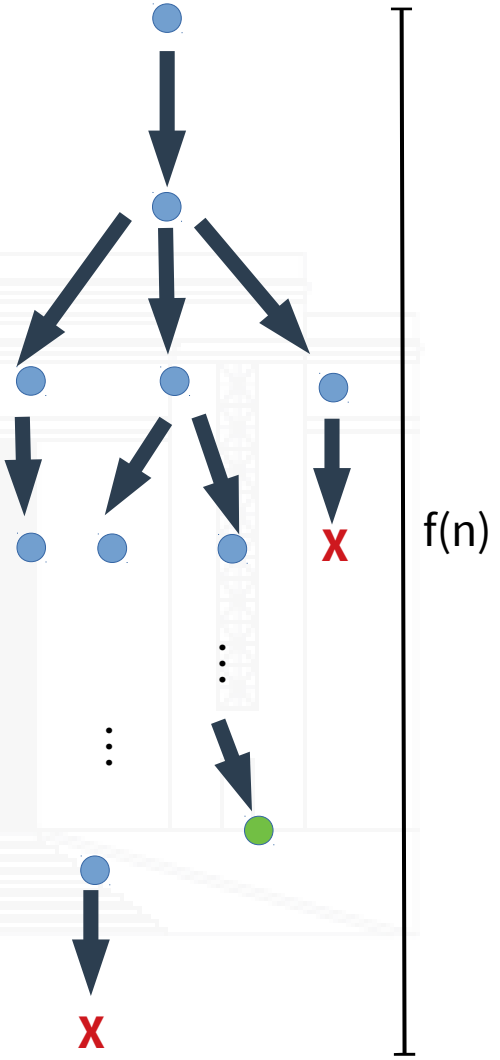
N una TM no deterministica decididora

El tiempo de ejecución de N

es una función f:N → N ,

Donde

f(n) es el número máximo de pasos que N realiza en alguna rama de su cómputo para cualquier input de tamaño n



Clase de Complejidad P

P corresponde a la clase de lenguajes

Que son decidibles en tiempo polinómico

Utilizando

Una máquina de turing deterministica con cinta única

$$P = \cup_k TIME\left(n^k\right)$$

Clase de Complejidad NP

P corresponde a la clase de lenguajes

Que son decidibles en tiempo polinómico

Utilizando

Una máquina de turing no deterministica con cinta única

Definimos

NTIME(t(n)) = {L| L es un lenguaje decidido por un O(t(n))-tiempo MT no determínisitica }

NP: nondeterministic polynomial time

$$NP = \cup_k NTIME(n^k)$$

Verificador

Un verificador

Para un lenguaje A es un algoritmo V

donde

A = {w| V acepta (w,c) para algun string c certificado}.

Medimos el tiempo del verificador en función del largo de w

Un verificador polinomial, se ejecuta en tiempo polinómico del largo de w

Un lenguaje A

Es polinomialmente verificable si tiene un verificador que se ejecuta en tiempo polinomial

Ejemplo

COMPUESTOS = {x| x = pq, con p, q > 1 enteros} ← p y q son los verificadores p*q = x el verificador

Clase de Complejidad NP (II)

Utilizando el concepto de verificador

Podemos definir la clase NP en forma alternativa

NP corresponde a la clase de lenguajes

Que tienen un verificador en tiempo polinomial

Es decir, que existe una TM determinística,

Que dado el input y el certificado,

decide si el input w pertenece al lenguaje utilizando el certificado

Función computable

Una función f :Σ* → Σ*

es una función computable

Si alguna Maquina de Turing M

Para cada input w, finaliza su ejecución con solo f(w) en su cinta

Las funciones computables pueden utilizarse

Para transformar descripción de maquinas

Por ejemplo

si w corresponde a M una maquina de turing

f(w) puede retornar M' la descripción de una nueva maquina de Turing

Reducción entre 2 lenguajes

El lenguaje A es mapeable reducible al lenguaje B

indicándolo $A \leq_m B$,

Si

Existe una función computable $f : \Sigma^* \rightarrow \Sigma^*$

Donde

Para cada w, $w \in A \Leftrightarrow f(w) \in B$

La función f

Es llamada la reducción de A a B

Reducción polinomial

Una función f :Σ* → Σ*

es una funcion computable en tiempo polinomial

Si alguna Maquina de Turing M en tiempo polinomial

Para cada input w, finaliza su ejecución con solo f(w) en su cinta

El lenguaje A es reducible polinomialmente al lenguaje B

indicándolo $A \leq_p B$,

Si Existe una función computable en tiempo polinomial f :Σ* → Σ*

Donde para cada w, $w \in A \Leftrightarrow f(w) \in B$

Clase de complejidad NP-Completo

Un lenguaje B

es NP-completo

Si satisface 2 condiciones

1. B pertenece a NP, y
2. todo A que peternece a NP se puede reducir en tiempo polinomial a B

Una vez que tenemos 1 problema NP-Completo

Podemos reducir polinomialmente C $\in$ NP-COMPLETO a B y evitar probarlo para todo problema en NP .

.UBAfiuba

FACULTAD DE INGENIERÍA

Presentación realizada en Julio de 2020