MÁQUINAS DE TURING II

COSTE COMPUTACIONAL

51

Bibliografía

Sipser – Introduction to the Theory of Computation – Chapter 3.

Ampliado en :

Alfonseca – Teoría de Lenguajes, Gramáticas y Autómatas, 2007 - Capítulo2.

y

Hopcroft, Motwani, Ullman – Introducción a la Teoría de Autómatas, Lenguajes, y Computación – Capítulo 8.

52

Coste Computacional con MT

En el pasado un uso que se dio a las MT fue el Análisis del Coste Computacional de los Algoritmos

Coste Computacional con MT

Uso para el Análisis del Coste Computacional de Algoritmos

Ventajas:

- *Propone un modelo único, al contrario que con el uso de un Lenguaje X, una Cpu Y, para cualquier X e Y*
- *No hay dudas sobre el coste del Paso Base*

Desventaja:

- *Es muy pesado para algoritmos complejos*
- *➔ Limitaremos el tamaño de las MT*

Coste Computacional con MT

Uso para Análisis del Coste Computacional de los Algoritmos

Propósito:

- *Relacionar conceptos vistos en la asignatura*
- *En Base a un Modelo muy sencillo*
- *Abstraer de construcciones de muy alto nivel (iteradores en Python)*

Coste Computacional con MT

Podemos aplicar los mismos conceptos vistos en Análisis de Coste Computacional:

- *Empírico*
- *Analítico*

Combinaremos ambos

Usaremos Jflap v7, en Jflap v6 funciona la traza, pero ambos tienen fallos.

- *Evitaremos usar Bloques según las circunstancias*

Coste Computacional con MT

Marco Necesario para el Análisis:

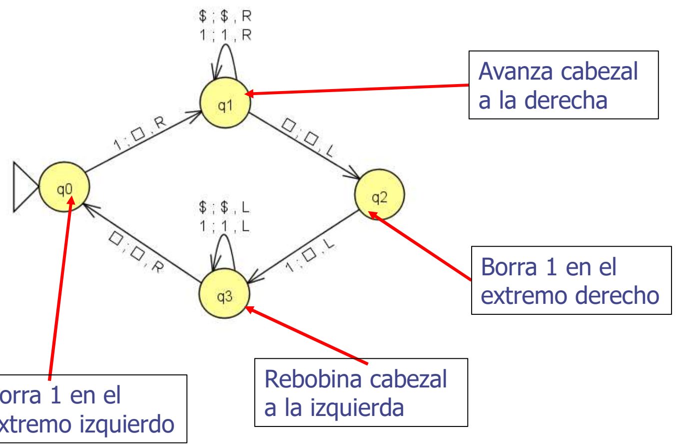
- *MT de cinta infinita hacia ambos lados*
- *Alfabeto determinado, en principio con pocos símbolos, pero pueden ser más*
- *Una Instancia de un Problema se describirá con una cinta de Entrada*
- *El número de símbolos de la Entrada define el Tamaño de la Instancia*
- *Coste de Paso Base: Una Transición con Lectura/Escritura y Desplazamiento*

MT para comparar dos números

- *Números en base 1*
- *Separados por $*

Cinta	#a$b#	
#1$1#	→	#1#
#11$11#	→	#1#
#111$111#	→	#1#
#$1#	→	#0#
#11$1#	→	#0#

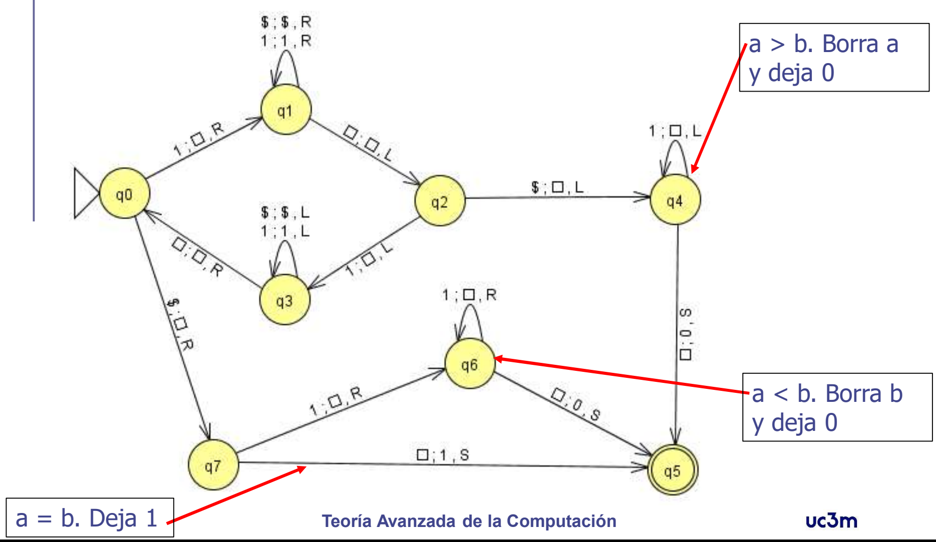
MT para comparar dos números



59

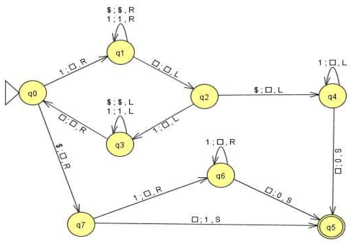
05 Modelos de Computación Máquinas de Turing II

MT para comparar dos números



60

MT para comparar dos números



Cinta de entrada:

###a$b###

Tamaño del problema n:

Tamaño de a + tamaño de b + tamaño de $

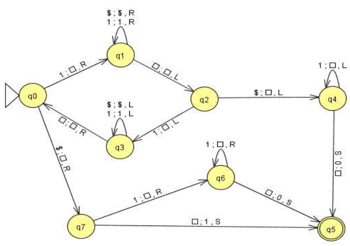
Tam(#1$1#) = 3

Tam(#11$11#) = 5

Tam(#$#) = 1

MT para comparar dos números

Coste computacional empírico:

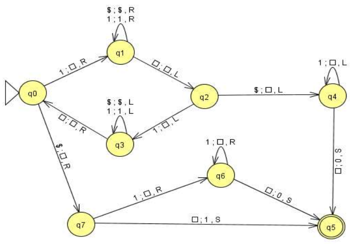


Tamaño n	1	3	5	7	9	11
Pasos	2	9	20	35	54	77

MT para comparar dos números

Coste computacional empírico:

Aplicamos diferencias finitas:



Tamaño n	1	3	5	7	9	11
Pasos	2	9	20	35	54	77
A: T(n)-T(n-2)		7	11	15	19	23
B: A(n)-A(n-2)		4	4	4	4	
C: B(n)-B(n-2)			0	0	0	

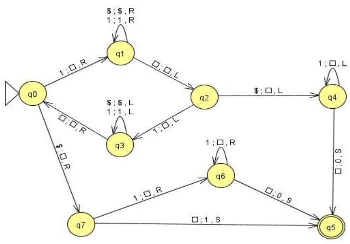
MT para comparar dos números

Coste computacional empírico:

Aplicamos diferencias finitas:

Se anulan en las de tercer orden

Son constantes en las de segundo orden

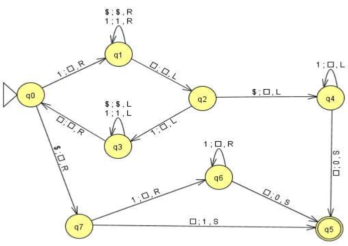


➔ Se puede aproximar con un polinomio de segundo orden

Tamaño n	1	3	5	7	9	11
Pasos	2	9	20	35	54	77
A: T(n)-T(n-2)		7	11	15	19	23
B: A(n)-A(n-2)		4	4	4	4	
C: B(n)-B(n-2)			0	0	0	

MT para comparar dos números

Coste computacional por diferencias finitas

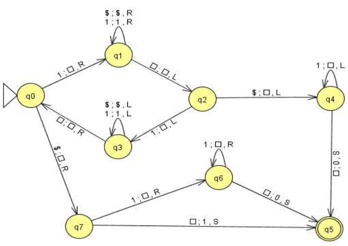


T(n) = an² + bn + c

Tamaño n	1	3	5	7	9	11
Pasos	2	9	20	35	54	77
A: T(n)-T(n-2)		7	11	15	19	23
B: A(n)-A(n-2)		4	4	4	4	
C: B(n)-B(n-2)			0	0	0	

MT para comparar dos números

Coste computacional por diferencias finitas



T(n) = an² + bn + c

T(1) = 2, T(3) = 9, T(5) = 20

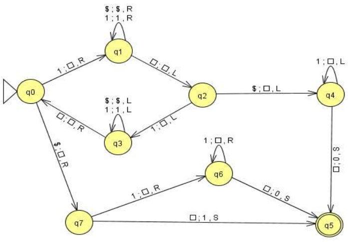
a+b+c=2 , 9a+3b+c=9 25a+5b+c=20

→ a=1/2, b=3/2, c=0 ➔ T(n) = 1/2n² + 3/2n

Tamaño n	1	3	5	7	9	11
Pasos	2	9	20	35	54	77
A: T(n)-T(n-2)		7	11	15	19	23
B: A(n)-A(n-2)		4	4	4	4	
C: B(n)-B(n-2)			0	0	0	

MT para comparar dos números

Otra forma de plantearlo (recursivamente)



T(n) en términos de T(n-2)

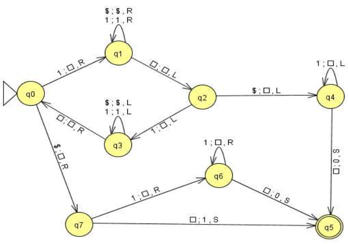
Para n=1 ➔ 2 pasos

Para n=3 ➔ 4 pasos a derecha + 3 a izquierda + T(1)

Para n=5 ➔ 6 pasos a derecha + 5 a izquierda + T(3)

MT para comparar dos números

Otra forma de plantearlo (recursivamente)



T(n) en términos de T(n-2)

Para n=1 ➔ 2 pasos

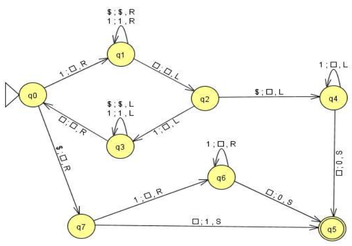
Para n=3 ➔ 4 pasos a derecha + 3 a izquierda + T(1)

Para n=5 ➔ 6 pasos a derecha + 5 a izquierda + T(3)

Para n➔ n+1 pasos a derecha + n a izquierda + T(n-2) =

= Σi para i=2…n+1 ➔ $T(n) = (n+1)(n+2)/2 - 1 = 1/2n^2 + 3n/2$

MT para comparar dos números



Otra forma de plantearlo (recursivamente)

$T(n) = 1/2n^2 + 3n/2 = \Sigma\, i$ para $i = 2 \ldots n+1$

Es un comportamiento habitual en MTs

que van marcando de extremo a extremo

Coste Computacional con MT

Ejemplo: ***Máquina para determinar si una palabra es palíndromo***

$\Sigma = \{ a, b \}$

Dada $x \in \Sigma^*$, $|x| = 2k$, $k = 0, 1, 2, 3, \ldots$

$\exists\, w \in \Sigma^* \mid x = w \cdot w^{-1}$

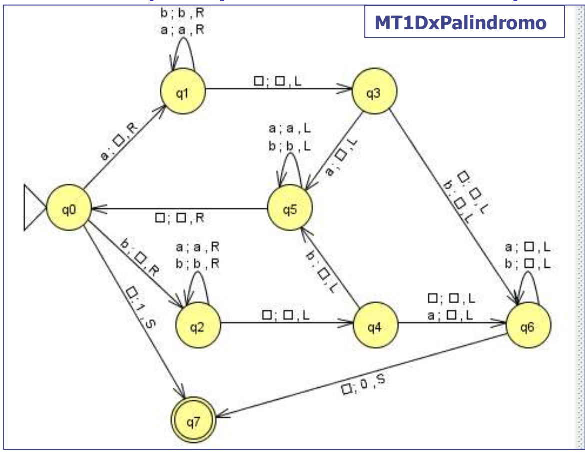
válidas: ***λ, aa, bb, aaaa, bbbb, abba, baab, aaaaaa, aabbaa, …***

inválidas: ***ab, ba, aaa, aba, bbb, abb, …, aabb, abab, baba, …***

Coste Computacional con MT

Ejemplo: **Máquina para determinar si una palabra es palíndromo**

MT1DxPalindromo

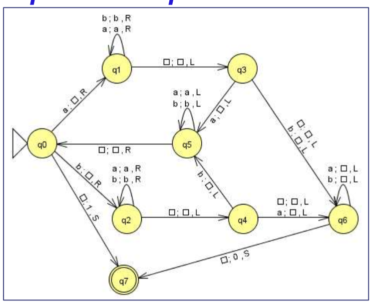


Input	Output	Result
a	0	Accept
aa	1	Accept
aaa	0	Accept
aaaa	1	Accept
aaaaa	0	Accept
aaaaaa	1	Accept
ababab	0	Accept
abbbba	1	Accept

71

Coste Computacional con MT

Ejemplo: **Máquina para determinar si una palabra es palíndromo**



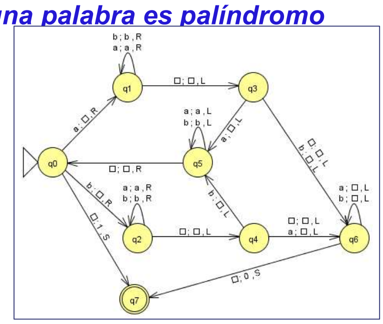
Funcionamiento:

```
Ciclo
   Borra una letra en extremo izqda.
   Busca la correspondiente a la dcha
   La borra
   Vuelve atrás a la primera letra por la izqda
Fin1:  Cinta en blanco ➔ Deja 1 en la cinta y Para (Acepta)
Fin2:  Falla una correspondencia ➔ Deja 1 en la cinta y Para ("Falla")
Fin3:  Incluye caso de número impar de letras ➔ se trata en Fin2
```


72

Coste Computacional con MT

Ejemplo: Máquina para determinar si una palabra es palíndromo

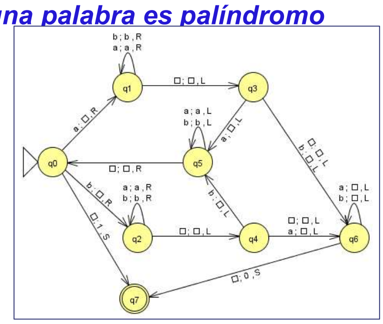


Detalles:

- Palabra de tamaño n=2k
- La MT hace k recorridos si n es el número de símbolos inicial
- Recorridos sucesivos se acortan en 2 símbolos
- En caso de fallar la correspondencia
  ➔ hay que borrar la cinta (q6) con un recorrido lineal
- El peor caso parece cuando es palíndromo para tamaños par (demostrar por contradicción)

Coste Computacional con MT

Ejemplo: Máquina para determinar si una palabra es palíndromo



Análisis: I. Fase Empírica:

Input	#tam	pasos
λ	0	1
aa	2	6
abba	4	15
aabaaba	6	28
aaabbaaa	8	45
bbaabbaabb	10	66
bbbbbaabbbbb	12	91

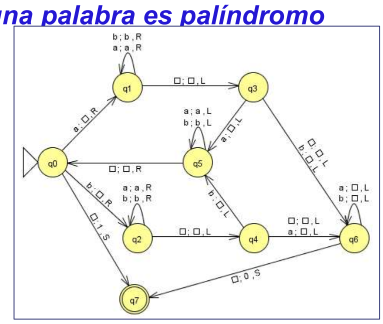
Coste Computacional con MT

Ejemplo: ***Máquina para determinar si una palabra es palíndromo***

Análisis: II. Fase Analítica:

Método de Diferencias Finitas:

N	0	2	4	6	8	10	12
Pasos	1	6	15	28	45	66	91
Dif 1		5	9	13	17	21	25
Dif 2		4	4	4	4	4	
Dif 3			0	0	0	0	



Constante

Al ser constantes las Diferencias Finitas segundas (y nulas las terceras) ➔ Podemos aproximar T(N) con polinomio de segundo orden

Coste Computacional con MT

Ejemplo: ***Máquina para determinar si una palabra es palíndromo***

Coste Empírico + Analítico

$T(N) = aN^2 + bN + c$

Método de Diferencias Finitas:

N	0	2	4	6	8	10	12
Pasos	1	6	15	28	45	66	91
Dif 1		5	9	13	17	21	25
Dif 2		4	4	4	4	4	
Dif 3			0	0	0	0	

Coste Computacional con MT

Ejemplo: *Máquina para determinar si una palabra es palíndromo*

Coste Empírico + Analítico

Método de Diferencias Finitas:

N	0	2	4	6	8	10	12
Pasos	1	6	15	28	45	66	91
Dif 1		5	9	13	17	21	25
Dif 2		4	4	4	4	4	
Dif 3			0	0	0	0	

$T(N) = aN^2 + bN + c$

$a \cdot 0^2 + b \cdot 0 + c = 1$
$a \cdot 2^2 + b \cdot 2 + c = 6$
$a \cdot 4^2 + b \cdot 4 + c = 15$

Coste Computacional con MT

Ejemplo: *Máquina para determinar si una palabra es palíndromo*

Coste Empírico + Analítico

Método de Diferencias Finitas:

N	0	2	4	6	8	10	12
Pasos	1	6	15	28	45	66	91
Dif 1		5	9	13	17	21	25
Dif 2		4	4	4	4	4	
Dif 3			0	0	0	0	

$T(N) = aN^2 + bN + c$

$a \cdot 0^2 + b \cdot 0 + c = 1 \quad \Rightarrow c = 1$
$a \cdot 2^2 + b \cdot 2 + c = 6$
$a \cdot 4^2 + b \cdot 4 + c = 15$

$4a + 2b = 5 \; (x2)$
$16a + 4b = 14$
\-----------------
$8a = 4 \quad \Rightarrow a = 1/2$

$4a + 2b = 5$
$2 + 2b = 5 \quad \Rightarrow b = 3/2$

Coste Computacional con MT

Ejemplo: Máquina para determinar si una palabra es palíndromo

Coste Empírico + Analítico

Método de Diferencias Finitas:

N	0	2	4	6	8	10	12
Pasos	1	6	15	28	45	66	91
Dif 1		5	9	13	17	21	25
Dif 2		4	4	4	4	4	
Dif 3			0	0	0	0	

$T(N) = aN^2 + bN + c$

$a\cdot0^2 + b\cdot0 + c = 1 \quad \Rightarrow c = 1$
$a\cdot2^2 + b\cdot2 + c = 6$
$a\cdot4^2 + b\cdot4 + c = 15$

$4a + 2b = 5$ (x2)
$16a + 4b = 14$

$8a = 4 \quad \Rightarrow a = 1/2$

$4a + 2b = 5$
$2 + 2b = 5 \quad \Rightarrow b = 3/2$

$T(N) = N^2/2 + 3N/2 + 1$

Coste Computacional con MT

Ejemplo: Máquina para determinar si una palabra es palíndromo

Análisis Recurrente (por Despliegue). Sabemos:

- Aplicar una iteración de MT para tamaño N reduce el problema al caso N-2
- Un recorrido para N símbolos supone 2N+1 pasos
- $T(0) = 1$ → Transición al estado final
- $T(2) = 2\cdot2+1 + T(0) = 2\cdot2+1 + 1 = 2\cdot2 + 2$
- $T(4) = 2\cdot4+1 + T(2) = 2\cdot4+1 + 2\cdot2 + 2 = 2\cdot4+2\cdot2 + 3 = 2\cdot6 + 3$
- $T(6) = 2\cdot6+1 + T(4) = 2.6+1 + 2\cdot6 + 3 = 2\cdot12 + 4$
- $T(8) = 2\cdot8+1 + T(6) = 2\cdot8+1 + 2\cdot12 + 4 = 2\cdot20+ 5$

$T(N) = N(N+2)/2 + N/2 + 1 = N^2/2 + 3N/2 + 1$

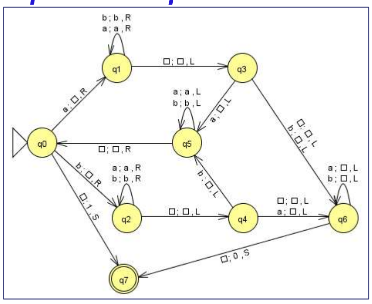
Coste Computacional con MT

Ejemplo: ***Máquina para determinar si una palabra es palíndromo***

Análisis de Recurrencia:

La MT opera:

iterando de extremo a extremo



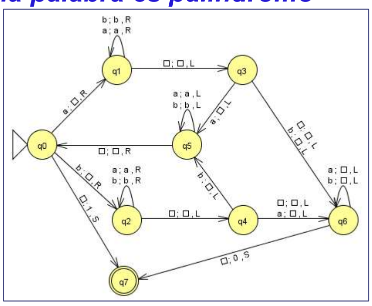
Es un caso interesante porque en cada iteración se reduce la instancia del problema a la instancia anterior

Coste Computacional con MT

Ejemplo: ***Máquina para determinar si una palabra es palíndromo***

TAREAS:

Optimizar esta MT para que aproveche el retroceso para cotejar más símbolos



Diferencias finitas

Dada una función T(n) cuya expresión analítica desconocemos

Disponemos de muestreos en T(n+a), T(n+b), etc.

Podemos trabajas con las diferencias finitas

T(n+a) - T(n+b)

Recordamos

(T(n+h) - T(n)) / h cuando $h \to \infty$ equivale a la derivada dT(n) / dn

Diferencias finitas

Dada una función T(n) cuya expresión analítica desconocemos

Disponemos de muestreos en T(n+a), T(n+b), etc.

Podemos trabajar con las diferencias finitas

Si T(n) es un polinomio de orden k

Si consideramos que n toma valores enteros

T(n+1) - T(n) sirve de aproximación a la primera derivada de T(n)

Diferencias finitas

Dada una función T(n) cuya expresión analítica desconocemos

Disponemos de muestreos en T(n+a), T(n+b), etc.

Podemos trabajar con las diferencias finitas

Esto da pie al Método de Diferencias Finitas

Se puede aplicar en Análisis Numérico a Ecuaciones Diferenciales

Nosotros lo usaremos a un nivel muy elemental.

Diferencias finitas

Dada una función T(n) cuya expresión analítica desconocemos

Disponemos de muestreos en T(n+a), T(n+b), etc.

Calculando diferencias finitas de primer, segundo, tercer orden

Podemos determinar el tipo de función (polinómica de orden k, exponencial)

Si T(n) polinomio de orden k, las diferencias finitas de orden k = cte

Si T(n) exponencial, las diferencias finitas no se anulan

Pero aparecen comportamientos particulares.

Diferencias finitas

Dada una función T(n) cuya expresión analítica desconocemos

Disponemos de muestreos en T(n+a), T(n+b), etc.

Calculando diferencias finitas de primer, segundo, tercer orden

Si T(n) = E(n) + P(n), E(n) exponencial, P(n) polinomio de orden k
aplicando k diferencias finitas ➔ anulamos polinomio

Podemos obtener E(n) mediante integración.