Diseño de algoritmos

Resolución de recurrencias

Jesús Bermúdez de Andrés

Universidad del País Vasco/Euskal Herriko Unibertsitatea **(UPV/EHU)**

Curso 2008-09

Análisis de algoritmos recursivos

Cuando analizamos algoritmos recursivos es útil describir la función de coste como una *recurrencia*.
Una *recurrencia* es una ecuación que describe una función en términos del propio valor de la función para argumentos más cercanos a algún caso básico, para el que la función está definida explícitamente.
Resolver una recurrencia significa encontrar la expresión explícita que define la función recurrente.

Generalmente, nos bastará con clasificar la función recurrente en una notación asintótica.

Resolución de recurrencias mediante la **técnica de expansión** (1/2)

Ejemplo: Dada la recurrencia

$$T(n) = \begin{cases} a & \text{si } n \leq 1 \\ T(n/2) + b & \text{si } n > 1 \end{cases}$$

- Sustituya cada aparición del valor de la función por la expresión especificada por la recurrencia.
  De $T(n/2) = T(n/2^2) + b$ obtenemos $T(n) = T(n/2^2) + b + b$
- Repita la sustitución correspondiente alguna vez más, hasta que sea capaz de "adivinar" un expresión patrón que generalice todas las sustituciones sucesivas.
  Así:
  $T(n) = T(n/2^2) + b + b = T(n/2^3) + b + b + b = \ldots = T(n/2^i) + ib$

Resolución de recurrencias mediante la **técnica de expansión** (2/2)

- Lleve el argumento de la función hasta algún caso básico y calcule así el valor de i.
  Así: $n/2^i = 1$ entonces $i = \lg_2 n$
- Sustituya i por su valor y obtendrá una expresión.
  Así: $T(n) = T(n/2^i) + ib = T(1) + b \lg_2 n = a + b \lg_2 n$
- Para ser rigurosos, debería demostrarse por inducción que esa es la solución.

Resolución de recurrencias lineales: **método de la ecuación característica** (1/2)

Las recurrencias lineales homogéneas son de la forma

$$a_n T(n) + a_{n-1} T(n-1) + \ldots + a_{n-k} T(n-k) = 0$$

y las recurrencias lineales no homogéneas de la forma

$$a_n T(n) + a_{n-1} T(n-1) + \ldots + a_{n-k} T(n-k) = b^n p(n)$$

siendo $p(n)$ un polinomio en n.

En ambos casos, lo primero es calcular las raíces de la *ecuación característica*

$$a_n x^k + a_{n-1} x^{k-1} + \ldots + a_{n-k} = 0$$

Resolución de recurrencias lineales: **método de la ecuación característica** (2/2)

Supongamos que las raíces son: $r_1, \ldots, r_k$.
Entonces la solución de la recurrencia es de la forma

$$T(n) = c_1 r_1^n + \ldots + c_k r_k^n$$

Para encontrar los valores de las constantes $c_1, \ldots, c_k$ hay que establecer un sistema de ecuaciones con k casos particulares.

Para las recurrencias no homogéneas, además aparece la raíz b (de la parte derecha de la ecuación) con multiplicidad $d + 1$ siendo d el grado del polinomio $p(n)$.
Cuando hay una raíz múltiple r_j, de multiplicidad m, entonces en la solución deben aparecer los sumandos siguientes:

$$\ldots + d_0 r_j^n + d_1 n r_j^n + d_2 n^2 r_j^n \ldots + d_{m-1} n^{m-1} r_j^n$$

Recurrencias típicas del análisis de algoritmos recursivos (1/2)

Teorema

Si

$$T(n) = a \cdot T(n-c) + \Theta(n^k)$$

siendo a, c y k constantes con $c \geq 1$ y $k \geq 0$, entonces

$$T(n) = \begin{cases} \Theta(n^k) & \text{si } a < 1 \\ \Theta(n^{k+1}) & \text{si } a = 1 \\ \Theta(a^{n/c}) & \text{si } a > 1 \end{cases}$$

Recurrencias típicas del análisis de algoritmos recursivos (2/2)

Teorema

Si

$$T(n) = a \cdot T(n/b) + \Theta(n^k)$$

siendo a, b y k constantes con $b > 1$ y $k \geq 0$, entonces

$$T(n) = \begin{cases} \Theta(n^k) & \text{si } a < b^k \\ \Theta(n^k \lg n) & \text{si } a = b^k \\ \Theta(n^{\lg_b a}) & \text{si } a > b^k \end{cases}$$