Apuntes de Teórico

PROGRAMACIÓN 3

Árboles de decisión

Versión 1.0

Índice

Introducción: árbol de decisión, árbol binario estricto

Un *Árbol de Decisión* (AD) es un **árbol binario estricto** (cada nodo tiene 2 hijos o ninguno) donde los **nodos internos tienen 2 hijos y los externos (hojas) no tienen ninguno**.

Estos árboles, vistos como grafos, son una herramienta para modelar la ejecución de algoritmos que resuelvan un problema dado. En este marco, los nodos internos representan operaciones y los externos representan el final de la ejecución de un algoritmo, cada uno para una determinada entrada.

Se adoptará como convención que si la comparación resulta verdadera, se toma a la derecha y de lo contrario a la izquierda.

Cualquier algoritmo se puede modelar con un AD particular. Una ejecución del algoritmo, para una determinada entrada, está representado por el camino de la raíz a un único nodo externo. El costo, en cantidad de operaciones para dicha entrada, estará dado por el largo de dicho camino (cantidad de aristas). De esta forma el peor caso corresponderá camino más largo de la raíz a un nodo externo.

Para el problema de encontrar cotas inferiores, se trabajará entonces con la herramienta árbol de decisión. De esta manera se logra un razonamiento genérico e independiente de cualquier algoritmo particular.

Árbol Binario Estricto

Un **árbol binario estricto** es un árbol (binario) donde los **nodos internos tienen 2 hijos y los externos (hojas) no tienen ninguno**.

Profundidad

Profundidad de un nodo i:

Prof(i) = Largo del camino (cantidad de aristas) desde la raíz al nodo ***i***.

notar que es igual a la cantidad de nodos internos.

Profundidad del Arbol:

Prof(T) = Máxima Profundidad de nodo externo

Propiedades de Árbol Binario Estricto (ABE)

Sea ***T*** un ABE

Sea ***m*** la cantidad de nodos externos de ***T*** (ABE)

Sea ***k*** la Profundidad de ***T***

1) Lema 1: $\mathbf{m \leq 2^k}$

No se demostrará formalmente pero por ser un ABE cada nodo tiene 2 hijos salvo las hojas que no tienen ninguno.

Si T es completo tendrá la máxima cantidad de nodos externos y en el nivel **k** esa cantidad será igual a $\mathbf{2^k}$, por lo tanto:

$\mathbf{m \leq 2^k}$

2) Lema 2: La suma de las profundidades de los ***m*** nodos externos es mayor o igual a ***mlog m:***

Sea D(T) la suma de las profundidades de los nodos externos:

$$D(T) = \sum_{j \text{ externo}} Prof(j)$$

Sea $\mathbf{d(m) = \min_{T} \{ D(T) / T \text{ tiene m nodos externos} \}}$

Demostración:

Se demostrará que **d(m) ≥ m log m**

como d(m) es el mínimo de los D(T), entonces quedará demostrado el lema.

Se usará *inducción completa* en m:

Paso base) m=1

d(1) es el mínimo de la suma de profundidades para todo T con 1 nodo externo, por lo tanto, por ser un ABE, T tiene un solo nodo, su profundidad será 0.

d(1) = 0

Por otro lado m log m = 1 log 1 = 0

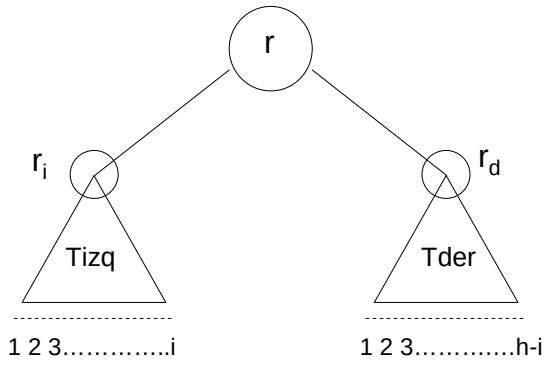
El paso base se cumple

HI) Si $1 \leq m < h \Rightarrow d(m) \geq m \log m$

TI) Si $m=h \Rightarrow d(h) \geq h \log h$

dem: Sea T el ABE considerado

h>1, T tiene 2 subárboles no vacíos (si no, no sería un ABE):



Sea ***i*** la cantidad de nodos externos del subárbol izquierdo T_{izq}, será ***h-i*** la cantidad de nodos externos del subárbol derecho T_{der}.

Como T_{izq} y T_{der} no son vacíos, serán

i < h y h-i < h

en ambos subárboles aplica la hipótesis inductiva:

d(i) ≥ i log i en el subárbol **T_{izq} de raíz r_i**

notar que:

- la profundidad está medida desde r_i
- d(i) = min {D(T) / T tiene **i** nodos externos }

d(h-i) ≥ (h-i) log (h-i) en el subárbol **T_{der} de raíz r_d**

notar que:

- la profundidad está medida desde r_d
- d(h-i) = min {D(T) / T tiene (h-i) nodos externos }

Para la demostración de la tésis inductiva se deberá "reconstruir" el árbol T y calcular **d(h)** midiendo la profundidad desde **r** (utilizando las conclusiones anteriores):

- Notar que se agrega la arista <**r,r_i** >en el lado izquierdo y la arista <**r,r_d** >en el derecho, además como se están sumando las profundidades de los nodos externos, esas aristas cuentan **i** veces del lado izquierdo y **h-i** veces del lado derecho
- Al descomponer T en dos subárboles y luego reconstruirlo deben considerarse todas las variaciones posibles **T_{izq}** y **T_{der}** . Los casos posibles varían desde que el subárbol izquierdo tenga 1 sólo nodo externo y el derecho h-2, luego 2 nodos externos en el izquierdo y h-3 en el derecho, hasta h-2 en el izquiero y 1 en el derecho (recordar que para T los subárboles no pueden ser vacíos).
- De las variantes anteriores se deberá tomar la que arroje un mínimo para **$D(T_{izq})$+ $D(T_{der})$.** De esta forma se asegurará que la fórmula siguiente se cumple para todas las combinaciones.

Por lo antedicho:

$d(h) \geq i + (h-i) + \min_{1 \leq i \leq h-1} \{ D(T_{izq}) + D(T_{der}) \}$

$$d(h) \geq h + \min_{1 \leq i \leq h-1} \{ i \log i + (h-i) \log (h-i) \}$$

El mínimo se da en i =h/2

d(h) ≥ h + 2(h/2 log h/2) = h + h(log h – log 2) = h + h log h – h = h log h

lqqd.