Lógica para ciencias de la computación Periodo: 2022-1
Taller: Recursión sobre números y árboles Profesor: E. Andrade

Sean n, m números naturales. Defina las funciones $\mathbf{2}\times[n]$, Pred$[n]$ y $\boldsymbol{m}-[n]$ de la siguiente manera:

función $\mathbf{2}\times[n]$:
  Si $n == 0$:
    retornar 0
  Si no:
    retornar $2 + \mathbf{2}\times[n-1]$

función $\textsc{Pred}[n]$:
  Si $n == 0$:
    retornar 0
  Si no:
    retornar $n-1$

función $\boldsymbol{m}-[n]$:
  Si $n == 0$:
    retornar m
  Si no:
    retornar $\textsc{Pred}[\boldsymbol{m}-[\textsc{Pred}[n]]]$

Ejercicio 1.

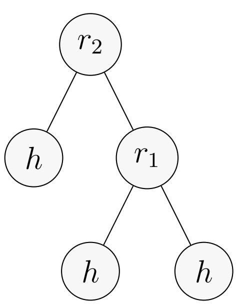
a) Escriba el paso a paso de $\mathbf{2}\times[3]$.

b) Escriba el paso a paso de $\boldsymbol{m}-[2]$ y de $\boldsymbol{m}-[3]$.

c) Demuestre por inducción sobre n que $\mathbf{2}\times[n]$ devuelve el número $2n$.

d) Suponga que m es un número natural arbitrario. Demuestre por inducción sobre n que $\boldsymbol{m}-[n]$ devuelve el número $\max\{0, m-n\}$ (es decir, el máximo entre 0 y el número $m-n$).

Una manera sencilla de escribir un árbol mediante la estructura `Tree`(left, right) es primero escribir sus subárboles y luego usarlos para construir el árbol de interés. Considere el siguiente árbol y su representación mediante la estructura `Tree`(left, right):



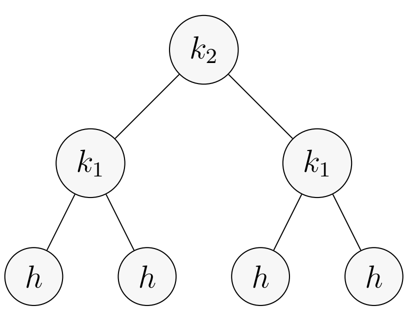
$h = \texttt{Tree}(\text{NULL}, \text{NULL})$

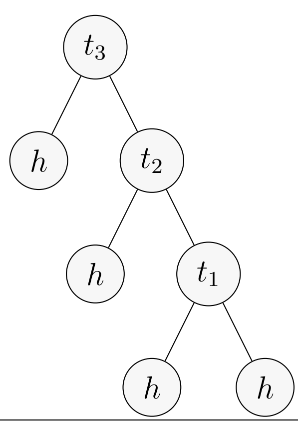
$r_1 = \texttt{Tree}(h, h)$

$r_2 = \texttt{Tree}(h, r_1)$

Observe que, por ejemplo, r_2.right=r_1 y que r_1.left=h

Ejercicio 2: Utilice la estructura `Tree`(left, right) para representar los siguientes árboles según el modelo dado en el ejemplo anterior:





Una función recursiva para contar el número de aristas de un árbol es la siguiente:

función num_aristas(A):
  Si A.right == NULL:
    retornar 0
  Si no:
    retornar 2 + num_aristas(A.left) + num_aristas(A.right)

El paso a paso de aplicar esta función al árbol r_2 es:

$$\begin{aligned}\text{num\_aristas}(r_2) &= 2 + \text{num\_aristas}(r_2.\text{left}) + \text{num\_aristas}(r_2.\text{right})\\ &= 2 + \text{num\_aristas}(h) + \text{num\_aristas}(r_1)\\ &= 2 + 0 + \text{num\_aristas}(r_1)\\ &= 2 + 0 + \big(2 + \text{num\_aristas}(r_1.\text{left}) + \text{num\_aristas}(r_1.\text{right})\big)\\ &= 2 + 0 + \big(2 + \text{num\_aristas}(h) + \text{num\_aristas}(h)\big)\\ &= 2 + 0 + (2 + 0 + 0) = 4\end{aligned}$$

Ejercicio 3: Presente el paso a paso de num_aristas para los árboles k_2 y t_3.

Una función recursiva para determinar la altura de un árbol es la siguiente:

función Altura(A):
  Si A.right == NULL:
    retornar 0
  Si no:
    retornar 1 + max{Altura(A.left), Altura(A.right)}

Ejercicio 4: Presente el paso a paso de Altura para los árboles r_2, k_2 y t_3.

Ejercicio 5: Defina una función recursiva num_nodos que encuentre el número de nodos de un árbol y presente el paso a paso de num_nodos para cada uno de los árboles del ejercicio 2.

Ejercicio 6: Demuestre mediante inducción estructural que, para cualquier árbol binario A:

$$\text{num\_nodos}(A) = \text{num\_aristas}(A) + 1$$

Ejercicio 7: Sea A un árbol binario y a=Altura(A). Demuestre por inducción estructural que:

$$\text{num\_nodos}(A) \leq 2^{a+1} - 1$$