Lógica para ciencias de la computación Periodo: 2019-1
Taller: Recursión sobre números y árboles Profesor: E. Andrade

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

```
DEF 2×[n]:
  SI n == 0:
    RETORNAR 0
  SI NO:
    RETORNAR 2 + 2×[n−1]
```

```
DEF PRED[n]:
  SI n == 0:
    RETORNAR 0
  SI NO:
    RETORNAR n−1
```

```
DEF m−[n]:
  SI n == 0:
    RETORNAR m
  SI NO:
    RETORNAR PRED[m−[PRED[n]]]
```

EJERCICIO 1.

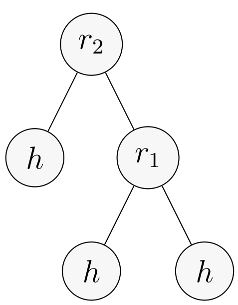
a) Escriba el paso a paso de $2\times[3]$.

b) Escriba el paso a paso de $3-[2]$ y de $2-[3]$.

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

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

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):



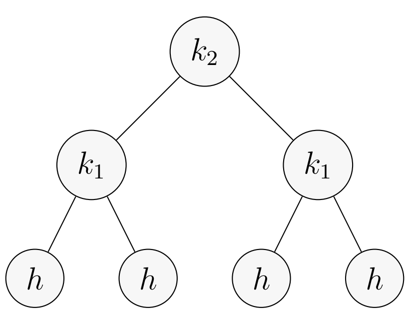
h = TREE(NULL, NULL)

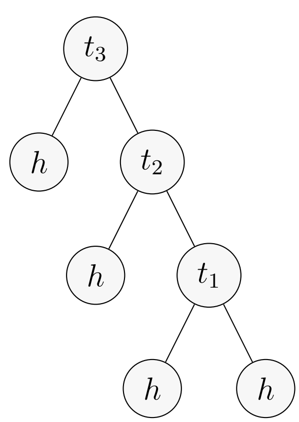
r_1 = TREE(h, h)

r_2 = 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 ramas de un árbol es la siguiente:

Def Num_Ramas(A):
  Si A.right == null:
    retornar 0
  Si no:
    retornar 2 + Num_Ramas(A.left) + Num_Ramas(A.right)

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

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

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

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

Def 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:

$$\textsc{Num\_Nodos}(A) = \textsc{Num\_Ramas}(A) + 1$$

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

$$\textsc{Num\_Nodos}(A) \leq 2^{a+1} - 1$$