Tarea Corta 3 - Adrián Amador Ávila

Trace de la función makeTree()

```
?- makeTree([3,4,5,1,2], [4,5,3,2,1], A).


makeTree(PreOrd, InOrd, Arbol) :-

      makeTree(PreOrd, InOrd, [], [], Arbol).


makeTree(A,B,A,B,null) :- !. (*)
makeTree([Raiz|Pre], In, PreOut, InOut, nodo(Raiz, Izq, Der)) :-
      makeTree(Pre, In, PreDer, [Raiz|InDer], Izq),
      makeTree(PreDer, InDer, PreOut, InOut, Der).
```

```
makeTree([3,4,5,1,2], [4,5,3,2,1], A).
```

Esto entraría en la primera función con 3 parámetros y haría una llamada a la función con 5 parámetros con los siguientes datos

```
makeTree(PreOrd, InOrd, Arbol) :-
      makeTree(PreOrd, InOrd, [], [], Arbol).

   -  Los siguientes datos representan los valores para las variables en la
      función

      PreOrd = [3,4,5,1,2], InOrd = [4,5,3,2,1], Arbol = no definido
```

Ahora evalúa la función con 5 parámetros, la cuál es la siguiente:

```
makeTree([Raiz|Pre], In, PreOut, InOut, nodo(Raiz, Izq, Der)) (**) :-
      makeTree(Pre, In, PreDer, [Raiz|InDer], Izq), {1}
      makeTree(PreDer, InDer, PreOut, InOut, Der). {2}

   -  El primer llamado a la función recursiva la vamos a llamar {1} y para
      la segunda llamada recursiva {2}, y para la llamada en general de
```

```
      toda la función (**) y la condición de parada (*)
```

Los datos presente en esta primer llamada a makeTree/5, se tiene los siguientes datos:

```
[Raiz|Pre] = [3|4,5,1,2]
In = [4,5,3,2,1]
PreOut = []
InOut = []
Nodo = (3, Izq, Der)
```

Ahora pasamos a la llamada de **{1}** con los siguientes datos para generar el nodo izquierdo, y la listas PreDer e InDer

```
{1-1}
makeTree(Pre, In, PreDer, [Raiz|InDer], Izq)
Pre = [4,5,1,2]
In = [4,5,3,2,1]
PreDer = por definir (->1)
[Raiz|InDer] = [3|por definir (->2)]
Izq = un nodo con su respectivo izq y der
```

Ahora volvemos a llamar a **(**)** con esos datos

```
{1-2}
makeTree([Raiz|Pre], In, PreOut, InOut, nodo(Raiz, Izq, Der)) :-
[Rainz|Pre] = [4|5,1,2]
In = [4,5,3,2,1]
PreOut = PreDer (->1)
InOut = [3| (->2)]
nodo = (4, Izq, Der)
```

Dentro de **(**)** se vuelve a realizar una llamada a **{1}** y se mostraría de la siguiente manera

```
{1-3}
makeTree(Pre, In, PreDer, [Raiz|InDer], Izq)
Pre = [5,1,2]
In = [4,5,3,2,1]
PreDer = por definir (->1.2)
[Raiz|InDer] = [4|por definir (->2.2)]
Izq = un nodo con su respectivo izq y der
```

Volvemos a llamar a **(**)** con esos datos

```
{1-4}
makeTree([Raiz|Pre], In, PreOut, InOut, nodo(Raiz, Izq, Der)) :-
[Rainz|Pre] = [5|1,2]
In = [4,5,3,2,1]
PreOut = PreDer (->1.2)
InOut = [4| (->2.2)]
nodo = (5, Izq, Der)
```

Dentro de **(**)** se vuelve a realizar una llamada a **{1}** con los siguientes datos

```
{1-5}
makeTree(Pre, In, PreDer, [Raiz|InDer], Izq)
Pre = [1,2]
In = [4,5,3,2,1]
PreDer = por definir (->1.3)
[Raiz|InDer] = [5|por definir (->2.3)]
Izq = un nodo con su respectivo izq y der
```

Volvemos a llamar a **(**)** con esos datos

```
{1-6}
makeTree([Raiz|Pre], In, PreOut, InOut, nodo(Raiz, Izq, Der)) :-
[Rainz|Pre] = [1|2]
In = [4,5,3,2,1]
PreOut = PreDer (->1.3)
InOut = [4| (->2.3)]
nodo = (1, Izq, Der)
```

Volvemos a realizar la llamada a **{1}** dentro de **(**)** con los siguientes datos

```
{1-7}
makeTree(Pre, In, PreDer, [Raiz|InDer], Izq)
Pre = [2]
In = [4,5,3,2,1]
PreDer = por definir (->1.4)
[Raiz|InDer] = [1|por definir (->2.4)]
Izq = un nodo con su respectivo izq y der
```

Se realiza la llamada a **(**)** con esos datos

```
{1-8}
makeTree([Raiz|Pre], In, PreOut, InOut, nodo(Raiz, Izq, Der)) :-
[Rainz|Pre] = [2| ]
In = [4,5,3,2,1]
PreOut = PreDer (->1.4)
InOut = [4| (->2.4)]
nodo = (2, Izq, Der)
```

Dentro de **(**)** se realiza la llamada a **{1}** con los siguientes datos

```
{1-9}
makeTree(Pre, In, PreDer, [Raiz|InDer], Izq)
Pre = []
In = [4,5,3,2,1]
PreDer = por definir (->1.5)
[Raiz|InDer] = [2|por definir (->2.5)]
Izq = un nodo con su respectivo izq y der
```

Para este caso en lugar de proceder con **(**)** se usa **(*)** ya que es la función que cumple para con los datos proporcionados

```
makeTree(A,B,A,B,null) :- !.
```

Acá, el tercer parámetro toma el valor del primero y el cuarto del segundo, además, el nodo tomaría el valor de null

```
A = []
B = [4,5,3,2,1]
null = nodo
```

Por lo tanto, para la llamada **{1-9}** tenemos que

```
PreDer = [] (->1.5)
[Raiz | InDer] = [2| (->2.5)] donde (->2.5) = [4,5,3,2,1] = [2|4,5,3,2,1]
Izq = null
```

Ahora, **{1-9}** sería el resultado para la llamada **{1-8}** y como ya terminó procede con **{2}** para lo cual se tendría

```
{2-1}
makeTree(PreDer, InDer, PreOut, InOut, Der)
PreDer = []
```

```
InDer = [4,5,3,2,1]
PreOut = por definir (->1.4)
InOut = [4| (->2.4)]
Der = (2, Izq, Der)
```

Como **{2-1}** no puede proceder por los datos que no sabe, se realiza un backtracking a **{1-7}**