Universidad Nacional Autónoma de México
Facultad de Ciencias
Estructuras Discretas
Tarea 3

Rubí Rojas Tania Michelle
taniarubi@ciencias.unam.mx
cuenta: 315121719

3 de noviembre de 2017

1. Demuestre que cada una de las siguientes fórmulas se cumple para cada $n \in \mathbb{N}$.

   a) $\sum_{i=1}^{n}(2i-1)^3 = n^2(2n^2-1)$

   Demostración. Inducción sobre n.

   - Base de inducción.
     n = 1. Este caso se cumple ya que
     $\sum_{i=1}^{1}(2i-1)^3 = (2(1)-1)^3 = (1)^3 = 1 = 1(2(1)-1) = (1)^2(2(1)^2-1)$
   - Hipótesis de inducción.
     Supongamos que el resultado es cierto para $n \geq 1$, es decir, supongamos que se cumple $\sum_{i=1}^{n}(2i-1)^3 = n^2(2n^2-1)$.
   - Paso inductivo.
     Tenemos que demostrar que la fórmula es válida para $n+1$, es decir, que se cumple $\sum_{i=1}^{n+1}(2i-1)^3 = (n+1)^2(2(n+1)^2-1) = (n+1)^2(2n^2+4n+1)$.
     Entonces

$$
\begin{aligned}
\sum_{i=1}^{n+1} &= \left(\sum_{i=1}^{n}(2i-1)^3\right) + (2(n+1)-1)^3 \\
&= (n^2(2n^2-1)) + (2(n+1)-1)^3 && \text{por H.I.} \\
&= (2^4 - n^2) + (8n^3 + 12n^2 + 6n + 1) && \text{desarrolando términos} \\
&= 2^4 + 8n^3 + 11n^2 + 6n + 1 && \text{agrupando términos semejantes} \\
&= (n+1)^2(2n^2+4n+1) && \text{factorizando}
\end{aligned}
$$

$\square$

   b) $\sum_{i=0}^{n} \frac{i}{2^i} = 2 - \frac{n+2}{2^n}$

   Demostración. Inducción sobre n.

   - Base de inducción.
     $n = 0$ Este caso se cumple ya que
     $\sum_{i=0}^{0} \frac{0}{2^0} = \frac{0}{1} = 0 = 2 - 2 = 2 - \frac{2}{1} = 2 - \frac{0+2}{2^0}$

- Hipótesis de inducción. Supongamos que el resultado es válido para $n \geq 0$, es decir, supongamos que se cumple $\sum_{i=0}^{n} \frac{i}{2^i} = 2 - \frac{n+2}{2^n}$.
- Paso inductivo.
  Tenemos que demostrar que la fórmula es válida para $n+1$, es decir, que se cumple $\sum_{i=0}^{n+1} \frac{i}{2^i} = 2 - \frac{(n+1)+2}{2^{n+1}} = 2 - \frac{n+3}{2^{n+1}}$
  Entonces

$$
\begin{aligned}
\sum_{i=0}^{n+1} \frac{i}{2^i} &= \left(\sum_{i=0}^{n} \frac{i}{2^i}\right) + \left(\frac{n+1}{2^{n+1}}\right) & \\
&= \left(2 - \frac{n+2}{2^n}\right) + \left(\frac{n+1}{2^{n+1}}\right) & \text{por H.I.} \\
&= 2 - \left(\frac{n+2}{2^n} + \frac{n+1}{2^{n+1}}\right) & \text{asociatividad} \\
&= 2 - \left(\frac{2(n+2) - (n+1)}{2^{n+1}}\right) & \text{resolviendo suma} \\
&= 2 - \left(\frac{2n+4-n+1}{2^{n+1}}\right) & \text{simplificando} \\
&= 2 - \left(\frac{n-3}{2^{n+1}}\right) & \text{simplificando}
\end{aligned}
$$

$\square$

2. Demuestre cada una de las siguientes desigualdades para los valores de $n \in \mathbb{N}$ especificados.

   a) $(1 + \frac{1}{n})^n < n$ para cada $n \in \mathbb{N}$ tal que $n \geq 3$

   Demostración. Inducción sobre n.

   - Base de inducción.
     $n = 3$. Este caso se cumple ya que $(1 + \frac{1}{3})^3 = (\frac{4}{3})^3 = \frac{64}{27} < 3$.
   - Hipótesis de inducción. Supongamos que el resultado es válido para $n \geq 3$, es decir, supongamos que se cumple $(1 + \frac{1}{n})^n < n$.
   - Paso inductivo.
     Tenemos que demostrar que la fórmula se cumple para $n+1$, es decir, que se cumple $(1 + \frac{1}{n+1})^{n+1} < n+1$.
     Sabemos que, por propiedades en los números reales, se cumple $n < n+1 \Rightarrow \frac{1}{n+1} < \frac{1}{n}$, y por lo tanto, $1 + \frac{1}{n+1} < 1 + \frac{1}{n}$. Además, si a y b son reales, se tiene que $a < b \Rightarrow a^n < b^n,\ n \in \mathbb{N}$. Podemos utilizar este resultado con la desigualdad $1 + \frac{1}{n+1} < 1 + \frac{1}{n}$, y desarrollar para utilizar nuestra hipótesis.
     Entonces

$$
\begin{aligned}
\left(1 + \frac{1}{n+1}\right)^{n+1} &< \left(1 + \frac{1}{n}\right)^{n+1} & \text{por la observación anterior} \\
&= \left(1 + \frac{1}{n}\right)^{n} \left(1 + \frac{1}{n}\right) & \text{descomponiendo la expresión anterior} \\
&< n(1 + \frac{1}{n}) & \text{por H.I.} \\
&= n + 1 & \text{simplificando}
\end{aligned}
$$

$\square$

b) $7n < 2^n$ para cada $n \in \mathbb{N}$ tal que $n \geq 6$

Demostración. Inducción sobre n.

- Base de inducción.
  $n = 6$. Este caso se cumple ya que $7(6) = 42 < 64 = 2^6$.
- Hipótesis de inducción. Supongamos que el resultado es válido para $n \geq 6$, es decir, supongamos que se cumple $7n < 2^n$.
- Paso inductivo.
  Tenemos que demostrar que la fórmula es válida para $n+1$, es decir, que se cumple $7(n+1) < 2^{n+1}$.
  Entonces

$$
\begin{aligned}
7(n+1) &= 7n + 7 && \text{simplificando} \\
&< 2^n + 7 && \text{por H.I.} \\
&< 2^n + 2^n && \text{ya que } 7 < 2^n \text{ con } n \geq 6 \\
&= 2 \cdot 2^n && \text{simplificando} \\
&= 2^{n+1} && \text{simplificando}
\end{aligned}
$$

□

3. Demuestre que

$$\prod_{i=2}^{n}(1 - \tfrac{1}{i^2}) = (1 - \tfrac{1}{2^2}) \times \cdots \times (1 - \tfrac{1}{n^2}) = \tfrac{n+1}{2n}$$

Para cada $n \in \mathbb{N}$ tal que $n \geq 2$.

Demostración. Inducción sobre n.

- Base de inducción.
  $n = 2$. Este caso se cumple ya que $\prod_{i=2}^{2}(1 - \frac{1}{i^2}) = 1 - \frac{1}{2^2} = (1 - \frac{1}{4}) = \frac{3}{4} = \frac{2+1}{2(2)}$
- Hipótesis de inducción. Supongamos que el resultado es válido para $n \geq 2$, es decir, supongamos que se cumple $\prod_{i=2}^{n}(1 - \frac{1}{i^2}) = \frac{n+1}{2n}$
- Paso inductivo.
  Tenemos que demostrar que la fórmula es válida para $n+1$, es decir, que se cumple $\prod_{i=2}^{n+1} = \frac{(n+1)+1}{2(n+1)} = \frac{n+2}{2n+2}$.

Entonces

$$\begin{aligned}
\prod_{i=2}^{n+1} &= \left(\prod_{i=2}^{n}\left(1-\frac{1}{i^2}\right)\right)\cdot\left(1-\frac{1}{(n+1)^2}\right) \\
&= \left(\frac{n+1}{2n}\right)\cdot\left(1-\frac{1}{(n+1)^2}\right) && \text{por H.I.} \\
&= \left(\frac{n+1}{2n}\right)\cdot\left(\frac{n^2+2n}{(n+1)^2}\right) && \text{resolviendo resta} \\
&= \frac{(n+1)(n^2+2n)}{2n(n+1)^2} && \text{resolviendo multiplicación} \\
&= \frac{n^2+2n}{2n(n+1)} && \text{eliminando el término } (n+1) \\
&= \frac{n(n+2)}{2n(n+1)} && \text{factorizando} \\
&= \frac{n+2}{2n+2} && \text{eliminando el término } n \text{ y simplificando}
\end{aligned}$$

$\square$

4. Sean $\{r_i\}_{i\in\mathbb{N}^\times}$ la sucesión definida por $r_1 = 1$, y $r_{n+1} = 4r_n + 7$ para cada $n \in \mathbb{N}$. Demuestre que $r_n = \frac{1}{3}(10\cdot 4^{n-1} - 7)$ para cada $n \in \mathbb{N}^\times$.

   Demostración. Inducción sobre n.

   - Base de inducción.
     $n = 1$. Este caso se cumple ya que $r_i = 1 = \frac{1}{3}(3) = \frac{1}{3}(10\cdot 1-7) = \frac{1}{3}(10\cdot 4^0-7) = \frac{1}{3}(10\cdot 4^{1-1}-7)$
   - Hipótesis de inducción. Supongamos que el resultado es válido para $n+1$, es decir, supongamos que se cumple $r_n = \frac{1}{3}(10\cdot 4^{n-1} - 7)$.
   - Paso inductivo.
     Tenemos que demostrar que la fórmula es válida para $n+1$, es decir, que se cumple $r_{r+1} = \frac{1}{3}(10\cdot 4^n - 7)$.
     Entonces

$$\begin{aligned}
r_{n+1} &= 4r_n + 7 && \text{definición recursiva de } r_{n+1} \\
&= 4(\frac{1}{3}(10\cdot 4^{n-1} - 7)) + 7 && \text{definición recursiva de } r_n \\
&= \frac{1}{3}(10\cdot 4^{n-1}\cdot 4 - 7\cdot 4) + 7 && \text{conmutatividad y simplificando} \\
&= \frac{1}{3}(10\cdot 4^n - 7(1+3)) + 7 && \text{simplificando y aplicando } 4 = 3+1 \\
&= \frac{1}{3}(10\cdot 4^n - 7 + 21) + 7 && \text{simplificando} \\
&= \frac{1}{3}(10\cdot 4^n - 7) - \frac{1}{3}\cdot 21 + 7 && \text{sacando al 21} \\
&= \frac{1}{3}(10\cdot 4^n - 7) - 7 + 7 && \text{resolviendo multiplicación} \\
&= \frac{1}{3}(10\cdot 4^n - 7) && \text{simplificando}
\end{aligned}$$

$\square$

5. Sean $\{b_i\}_{i\in\mathbb{N}}$ la sucesión definida por $d_0 = 2, d_1 = 3$, y $d_n = d_{n-1} \cdot d_{n-2}$ para cada $n \in \mathbb{N}$ tal que $n \geq 3$. Encuentre una fórmula explícita para d_n, y demuestre por inducción que su fórmula funciona.

Demostración. La fórmula explícita propuesta es $d_n = 2^{F_{n-1}} \cdot 3^{F_n}$, con $n \geq 1$ y donde F_n es el n-ésimo número de Fibonacci.

Demostraremos que la fórmula es válida utilizando inducción fuerte sobre n.

- Base de inducción.
  $n = 1$. Este caso se cumple ya que $d_1 = 3 = 1 \cdot 3 = 2^0 \cdot 3^1 = 2^{F_0} \cdot 3^{F_1} = 2^{F_1-1} \cdot 3^{F_1}$
- Hipótesis de inducción. Supongamos que el resultado es válido para n, es decir, supongamos que se cumple para $d_n = 2^{F_{n-1}} \cdot 3^{F_n}$.
- Paso inductivo.
  Tenemos que demostrar que la fórmula es válida para $n + 1$, es decir, que se cumple $d_{n+1} = 2^{F_n} \cdot 3^{F_{n+1}}$.
  Entonces

$$
\begin{aligned}
d_{n+1} &= d_{(n+1)-1} \cdot d_{(n+1)-2} && \text{definición recursiva de } d_n \\
&= d_n \cdot d_{n-1} && \text{simplificando} \\
&= 2^{F_{n-1}} \cdot 3^{F_n} \cdot 2^{F_{(n-1)-1}} \cdot 3^{F_{n-1}} && \text{por H.I.} \\
&= 2^{F_{n-1}} \cdot 2^{F_{n-2}} \cdot 3^{F_n} \cdot 3^{F_{n-1}} && \text{simplificando y aplicando conmutatividad} \\
&= 2^{F_{n-1}+F_{n-2}} \cdot 3^{F_n+F_{n-1}} && \text{simplificando} \\
&= 2^{F_n} \cdot 3^{F_{n+1}} && \text{definición de } F_n \text{ y } F_{n+1}
\end{aligned}
$$

□

6. Sea *spar(n)* la función definida como $span(n) = 2 + 4 + 6 + \cdots + 2n$. Defina una implementación recursiva llamada $f(n)$ para la función $spar(n)$. Demuestre que $f(n) = n(n + 1)$.

Demostración. Definimos la función como $f(0) = 0, f(n+1) = f(n)+2(n+1)$. Demostraremos, por inducción sobre n, que $f(n) = n(n + 1)$.

- Base de inducción.
  $n = 0$. Este caso se cumple ya que $f(0) = 0 = 0(1) = 0(0 + 1)$.
- Hipótesis de inducción. Supongamos que el resultado es válido para n, es decir, supongamos que se cumple $f(n) = n(n + 1)$.
- Paso inductivo.
  Tenemos que demostrar que la fórmula es válida para $n+1$, es decir, que se cumple $f(n+1) = (n + 1)(n + 2)$.
  Entonces

$$
\begin{aligned}
f(n+1) &= f(n) + 2(n+1) && \text{definición recursiva de } f(n+1) \\
&= n(n+1) + 2(n+1) && \text{por H.I.} \\
&= n^2 + n + 2n + 2 && \text{simplificando} \\
&= n^2 + 3n + 2 && \text{agrupando términos semejantes} \\
&= (n+1)(n+2) && \text{factorizando}
\end{aligned}
$$

□

7. Una cadena de caracteres es palíndroma si es de la forma ww^R donde w^R es w escrita de atrás hacia adelante, por ejemplo, $0110, abbbba, holaaloh$. Defina al conjunto de las cadenas palíndromas recursivamente, y demuestre mediante inducción estructural, que todas las cadenas palíndromas definidas tienen un número par de símbolos.

   Demostración. Sea $\sum$ el alfabeto sobre el cual construiremos a los palíndromos. Definimos al conjunto P de las cadenas palíndromas de la siguiente forma

   i) Para cada $a \in \sum$, $aa^R \in P$.
   ii) Si v y w son cadenas tales que vv^R y ww^R son elementos de P, entonces también wvv^Rw^R lo es.
   iii) Sólo las cadenas obtenidas con las reglas $i)$ y $ii)$ son elementos de P.

   Ahoram demostraremos que todos los elementos del conjunto P tienen un número par de símbolos, utilizando inducción estructural.

   - Base de inducción.
     Cualquier cadena en P construida con la regla $i)$ es de la forma aa^R para algún símbolo en $\sum$, por lo que tiene exactamente dos elementos.
   - Hipótesis de inducción. Supongamos que uu^R y vv^R son cadenas en P, y que ambas tienen un número par de símbolos.
   - Paso inductivo.
     Demostraremos que la cadena uvv^Ru^R se puede obtener con la regla $ii)$ a partir de las cadenas u y v, que tiene un número par de símbolos. Como el número de símbolos uvv^Ru^R es la suma del número de símbolos en uu^R y vv^R, por la hipótesis tenemos que éstas cadenas tienen longitud par; y como la suma de un par con otro par es un par, podemos concluir que el resultado deseado.

   □

8. La función *snoc* en listas se define como sigue:

$$snoc \;\; c[x_1, \cdots, x_n] = [x_1, \cdots, x_n, c]$$

   a) De una implementación recursiva para *snoc*.
      SOLUCIÓN: Definimos la función *snoc* de la siguiente forma:

$$snoc\ c\ [] = [c]$$
$$snoc\ c\ (a : xs) = (a : snoc\ c\ xs)$$

   b) Demuestre, usando la definición recursiva, que:

$$snoc \;\; c\ (xs\_ys) = xs\_(snoc \;\; c\ ys)$$

   Demostración. Inducción estructural sobre xs.

   - Base de inducción.
     $xs = []$. Este caso se cumple ya que $snoc\ c\ ([]\_ys) = snoc\ c\ ys = []\_(snoc\ c\ ys)$
   - Hipótesis de inducción. Supongamos que se cumple $snoc \;\; c\ (xs\_ys) = xs\_(snoc \;\; c\ ys)$.

- Paso inductivo.
  Debemos demostrar que para cualquier a, se cumple que $snoc\ c\ ((a : xs)\_ys) = (a : xs)\_(snoc\ \ c\, ys)$.
  Entonces

$$\begin{aligned}
snoc\, c\, ((a : xs)_ys) &= snoc\, c\, (a : (xs_ys)) && \text{asociatividad de la concatenación}\\
&= (a : snoc\, c\, (xs_ys)) && \text{definición recursiva de } snoc\\
&= (a : xs_(snoc\, c\, ys)) && \text{por H.I.}\\
&= (a : xs)_(snoc\, c\, ys) && \text{asociatividad de la concatenación}
\end{aligned}$$

□

9. Considere la siguiente función misteriosa *mist*

$$\begin{gathered}
mist\ []\ ys = ys\\
mist\ (x : xs)\ ys = mist\ xs\ (x : ys)
\end{gathered}$$

   a) ¿Qué hace la función *mist*?
      SOLUCIÓN: La función *mist* recibe dos listas xs y ys, y concatena la reversa de xs con ys.
   b) Muestre que $rev\ xs = mist\ xs\ []$, con rev la operación reversa sobre cadenas definidas cómo sigue:

$$\begin{gathered}
rev\ [] = []\\
rev\ (a : xs) = rev\ xs_[a]
\end{gathered}$$

Demostración. Inducción estructural sobre xs.

- Base de inducción.
  $xs = []$. Este caso se cumple ya que $rev\ [] = [] = mist\ []\ []$.
- Hipótesis de inducción. Supongamos que se cumple $rev\ xs = mist\ xs\ []$.
- Tenemos que demostrar que para cualquier a, se cumple $rev\ (a : xs) = mist\ (a : xs)\ []$.
  Entonces

$$\begin{aligned}
rev\ (a : xs) &= rev\ xs_[a] && \text{definición recursiva de } rev\\
&= (mist\ xs\ [])_[a] && \text{por H.I.}\\
&= mist\ xs\ (a : []) && \text{propiedades de } []\\
&= mist\ (a : xs)\ [] && \text{definición recursiva de } mist
\end{aligned}$$

□

10. Sea A una fórmula de la lógica proposicional cuyos únicos conectivos son $\wedge, \vee \neg$. Definimos la fórmula dual de A, denotada como A_D, intercambiando $\wedge$ por $\vee$, $\vee$ por $\wedge$ y reemplazando a cada variable p por su negación $\neg p$. Por ejemplo, $A = (r \vee q) \wedge \neg p$, $A_D = (\neg r \wedge \neg q) \vee \neg\neg p$.

   a) Defina recursivamente una función dual tal que $dual(A) = A_D$.
      SOLUCIÓN: Definimos la función $dual(A)$ de la siguiente forma
      - $dual(p) = \neg p$
      - $dual(true) = false$
      - $dual(false) = true$
      - $dual(\neg A) = \neg dual(A)$

- $dual(A \land B) = dual(A) \lor dual(B)$
- $dual(A \lor B) = dual(A) \land dual(B)$.

b) Muestre que $\neg A \equiv A_D$ mediante inducción sobre fórmulas.

Demostración. Inducción sobre las fórmulas.

- Base de inducción.
  A es atómica (no tiene operadores). Este caso se cumple, ya que, por la reflexividad de $\equiv$, tenemos que
  - $A = p \Rightarrow dual(p) = \neg p \equiv \neg p$
  - $A = true \Rightarrow dual(true) = false \equiv false$
  - $A = false \Rightarrow dual(false) = true \equiv true$
- Hipótesis de inducción. Supongamos que $\neg A \equiv A_D$ y $\neg B \equiv B_D$.
- Paso inductivo.
  Tenemos que
  - $A = \neg A \Rightarrow dual(\neg A) = \neg dual(A) \equiv \neg\neg A$
  - $A = A \land B \Rightarrow dual(A \land B) = dual(A) \lor dual(B) \equiv \neg A \lor \neg B \equiv \neg(A \land B)$
  - $A = A \lor B \Rightarrow dual(A \lor B) = dual(A) \land dual(B) \equiv \neg A \land \neg B \equiv \neg(A \lor B)$

$\square$

11. Resuelva los siguientes incisos para árboles binarios.

- Defina recursivamente una función $hmi(T)$ que devuelve la hoja más a la izquierda en un árbol binario.
  SOLUCIÓN: Definimos recursivamente la función $hmi(T)$ de la siguiente forma (suponiendo que nuestro árbol es diferente de *void*):
  - $hmi(tree(void, c, void)) = c$
  - $hmi(tree(T_1, c, T_2)) = hmi(T_1)$
  - $hmi(tree(T_1, c, void)) = hmi(T_1)$
  - $hmi(tree(void, c, T_2)) = hmi(T_2)$
- La distancia entre la raíz r de un árbol binario T hacía algún otro nodo p es el número de aristas (líneas) que hay entre ambos nodos y la altura o profundidad de un árbol se define como la máxima distancia entre la raíz y alguna hoja más 1. Demuestre que el número máximo de hojas en un árbol de altura n es 2^{n-1}.

  Demostración. Inducción estructural sobre T.

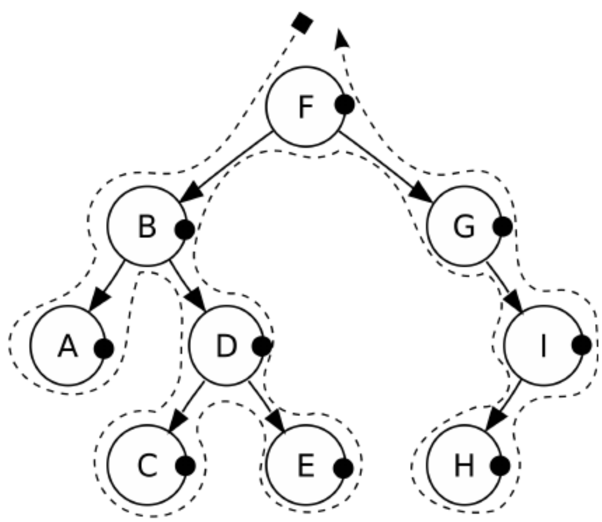
  - Base de inducción.
    $T = void$. Este caso se cumple ya que la altura y el número de hojas de T son iguales a 0. Así, $0 < \frac{1}{2} = 2^{-1} = 2^{0-1}$
  - Hipótesis de inducción. Si T es un árbol de altura n_i, entonces tiene a lo más 2^{n_i-1} hojas, donde $i \in \{1, 2\}$.
  - Paso inductivo.
    Sea $T = tree(T_1, c, T_2)$. Notemos que la altura de T es igual a $1 + max\{n_1, n_2\}$, por lo que debemos demostrar que el número de hojas de T es a lo más $2^{1+max\{n_1,n_2\}} = 2^{max\{n_1,n_2\}}$.
    Como el número de hojas de T es el número de hojas de T_1 más el número de hojas de T_2, entonces podemos definir recursivamente una función nh que calcule el número de hojas de

T. Así,

$$
\begin{aligned}
nh(T) &= nh(T_1) + nh(T_2) && \text{definición recursiva de } nh \\
&\leq 2^{n_1-1} + 2^{n_2-1} && \text{por H.I.} \\
&\leq 2^{max\{n_1,n_2\}-1} + 2^{max\{n_1,n_2\}-1} && \text{ya que } n_i = max\{n_1, n_2\} \\
&= 2 \cdot 2^{max\{n_1,n_2\}-1} && \text{aritmética} \\
&= 2^{max\{n_1,n_2\}-1+1} && \text{leyes de exponentes} \\
&= 2^{max\{n_1,n_2\}} && \text{simplificando}
\end{aligned}
$$

□

- De una definición recursiva que devuelva en una lista el recorrido post-orden de los árboles binarios. Si se tiene el siguiente árbol T, el resultado del recorrido es el siguiente:



$$\text{post-order}(T) = [A, C, E, D, B, H, I, G, F]$$

Solución: Definimos recursivamente la función $post - order$ de la siguiente forma

- $post - order(void) = []$
- $post - order(tree(T_1, c, T_2)) = post - order(T_1)\_post - order(T_2)\_[c]$