Considere la fórmula $p \vee (q \to r)$. Esta fórmula puede representarse mediante el árbol A_4 que se muestra a continuación:

$A_0 = \textsc{Tree}(r,\ \textsc{null},\ \textsc{null})$

$A_1 = \textsc{Tree}(q,\ \textsc{null},\ \textsc{null})$

$A_2 = \textsc{Tree}(\to,\ A_1,\ A_0)$

$A_3 = \textsc{Tree}(p,\ \textsc{null},\ \textsc{null})$

$A_4 = \textsc{Tree}(\vee,\ A_3,\ A_2)$

Ejercicio 1: Utilice la estructura Tree(label, left, right) para definir los árboles de las siguientes fórmulas según el modelo dado en el ejemplo anterior:

a. $p \wedge \neg q$

b. $\neg p \to \big((p \wedge \neg q) \to (p \wedge q)\big)$

c. $(\neg p \vee q) \wedge \Big(q \to \big((\neg r \wedge \neg p) \wedge (p \vee r)\big)\Big)$

Sea A una fórmula. La función recursiva que busca el número de ocurrencias de conectivos (incluida la negación) de A, Num_Conec, es la siguiente:

```
Def Num_Conec(A):
    Si A.right == null:
        retornar 0
    Si no, si A.label == ¬:
        retornar 1 + Num_Conec(A.right)
    Si no, si A.label ∈ {∧, ∨, →, ↔}:
        retornar 1 + Num_Conec(A.left) + Num_Conec(A.right)
```

El paso a paso de aplicar esta función a la fórmula $(p \vee (q \to r))$, representada mediante el árbol A_4, es:

$$\begin{aligned}
\textsc{Num_Conec}(A_4) &= 1 + \textsc{Num_Conec}(A_3) + \textsc{Num_Conec}(A_2)\\
&= 1 + 0 + \big(1 + \textsc{Num_Conec}(A_1) + \textsc{Num_Conec}(A_0)\big)\\
&= 1 + 0 + (1 + 0 + 0)
\end{aligned}$$

Ejercicio 2: Presente el paso a paso de Num_Conec para cada una de las fórmulas del ejercicio 1.

La función recursiva que convierte un árbol en una cadena de símbolos, Inorder, es la siguiente:

```
Def Inorder(A):
    Si A.right == null:
        Retornar A.label
    Si no, si A.label == ¬:
        Retornar "¬" + Inorder(A.right)
    Si no, si A.label ∈ {∧, ∨, →, ↔}:
        Retornar "(" + Inorder(A.left)
                 + A.label + Inorder(A.right) + ")"
```

Considere los siguientes árboles:

$A_0 = \textsc{Tree}(p,\ \textsc{null},\ \textsc{null})$

$A_1 = \textsc{Tree}(q,\ \textsc{null},\ \textsc{null})$

$A_2 = \textsc{Tree}(\neg,\ \textsc{null},\ A_0)$

$A_3 = \textsc{Tree}(\neg,\ \textsc{null},\ A_1)$

$A_4 = \textsc{Tree}(\rightarrow,\ A_0,\ A_1)$

$A_5 = \textsc{Tree}(\rightarrow,\ A_3,\ A_2)$

$A_6 = \textsc{Tree}(\leftrightarrow,\ A_4,\ A_5)$

Ejercicio 3: Presente el paso a paso de $\textsc{Inorder}(A_6)$.

Ejercicio 4: Defina de manera recursiva las funciones $P[A]$, la cual cuenta el número de ocurrencias de letras proposicionales de A, y $C[A]$, la cual cuenta el número de ocurrencias de conectivos binarios de A.

Nota: Observe que, por ejemplo, $P[(p \wedge \neg p) \wedge q] = 3$ y $C[(p \wedge \neg p) \wedge q] = 2$.

Ejercicio 5: Presente el paso a paso de las funciones definidas en el ejercicio 4 aplicada a cada una de las fórmulas del ejercicio 1.

Ejercicio 6: Demuestre por inducción estructural sobre A que $P[A] = C[A] + 1$.

Ejercicio 7: Escriba el paso a paso de

$$\mathrm{Sust}\Big[\neg((p \wedge q) \rightarrow (p \vee q)),\quad p \vee q,\quad \neg(\neg p \wedge \neg q)\Big]$$

Ejercicio 8: Sea B una fórmula y $A \in \mathrm{Subforms}(B)$. Sea A' una fórmula. Use la definición de $\mathrm{Sust}[B, A, A']$ para demostrar que:

$$\neg B\{A \leftarrow A'\} = \neg(B\{A \leftarrow A'\})$$

Nota: Observe que en la izquierda primero actúa la negación y luego Sust; en la derecha primero actúa Sust y luego la negación.

Ejercicio 9: Sea A una subfórmula de $B \odot C$. Sea A' una fórmula y $\odot \in \{\wedge, \vee \rightarrow, \leftrightarrow\}$. Use la definición de $\mathrm{Sust}[B, A, A']$ para demostrar que:

$$(B \odot C)\{A \leftarrow A'\} = B\{A \leftarrow A'\} \odot C\{A \leftarrow A'\}$$