

El objetivo de este taller es implementar el algoritmo de construcción de tableros semánticos (tableaux), el cual recibe una fórmula como árbol y devuelve una lista con las hojas abiertas del tableau.

Este taller consta de cuatro partes:

1. En la primera, cada estudiante de manera individual se familiarizará con la herramienta de Github para insertar código. [Duración: 15min]
2. En la segunda, los estudiantes se reunirán por parejas y revizarán la implementación del algoritmo de tableaux en Python, para culminar en una división de tareas de programación entre los dos estudiantes. [Duración: 30min]
3. En la tercera, cada estudiante realizará su respectiva tarea de programación y resolverán los ejercicios del 1 al 5. [Duración: 45min]
4. Después, en la cuarta parte, los estudiantes volverán a reunirse en las parejas ya establecidas, verificarán la ejecución satisfactoria del código, y lo utilizarán para realizar los ejercicios 6 y 7. Finalmente, resolverán el quiz virtual “Quiz Tableaux” en e-aulas [Duración: 20min]

## Parte 1

Uno de los propósitos de este taller es fomentar las habilidades de programación colaborativa mediante las herramientas proporcionadas por Github. Para ello, el estudiante debe estar familiarizado con la sincronización de un directorio local y un repositorio de Github, así como con la secuencia de comandos ADD, COMMIT, PUSH para insertar código en los archivos del repositorio.<sup>1</sup> Este es el tema que revisaremos en la Parte 1.

**Importante:** Antes de comenzar, asegúrese de tener creada una cuenta en Github y de crear un repositorio nuevo con el nombre Tableaux.

## Clonación del repositorio en un directorio local

Por favor siga cada uno de los pasos descritos a continuación.

1. En su navegador favorito de internet, por favor ingrese a su repositorio de Github y copie la url que aparece en el cuadro de texto superior (p.ej., <https://github.com/Slendercoder/Tableaux>).
2. Abra una ventana de terminal e introduzca el comando `pwd` para tomar nota del directorio actual (por ejemplo `hipatia/Downloads`).

<sup>1</sup>La explicación sobre Github que proporcionaremos en este taller es muy limitada. Para tener una visión más completa de las funcionalidades de Github puede consultarse, entre otros, <https://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1/> o <https://rogerdudler.github.io/git-guide/>.

3. En la ventana de terminal introduzca la siguiente instrucción:<sup>2</sup>

```
> git clone REPOSITORIO
```

Donde REPOSITORIO corresponde a la url copiada en el punto anterior. Por ejemplo:

```
> git clone https://github.com/Slendercoder/Tableaux
```

4. Ingrese al e-aulas del curso y busque en la pestaña de talleres la carpeta “Archivos taller implementacion tableaux”. Descargue los archivos `main.py` y `tableaux.py`. Guárdelos en el directorio clonado en el punto anterior (p.ej., `hipatia/Downloads/Tableaux`).
5. En la ventana de terminal, introduzca los comandos siguientes:

```
> cd Tableaux  
> git add main.py  
> git add tableaux.py  
> git commit -m 'Inicializando taller'  
> git push
```

El primer comando lo debe llevar al directorio recién clonado. El segundo le dice a Github que se desea añadir el archivo `main.py` al repositorio. El tercero comando hace lo propio con el archivo `tableaux.py`. El cuarto comando crea un mensaje que aparecerá en el repositorio en frente de los archivos añadidos. Este mensaje indica el punto de avance en que va el proyecto y las modificaciones incluidas. El último comando le dice a Github que incorpore los cambios introducidos en el repositorio que se encuentra en la nube.

**Importante:** Es muy probable que aparezca un error después de introducir el último comando. Verifique que dicho error le está indicando introducir sus credenciales (nombre de usuario y luego contraseña).

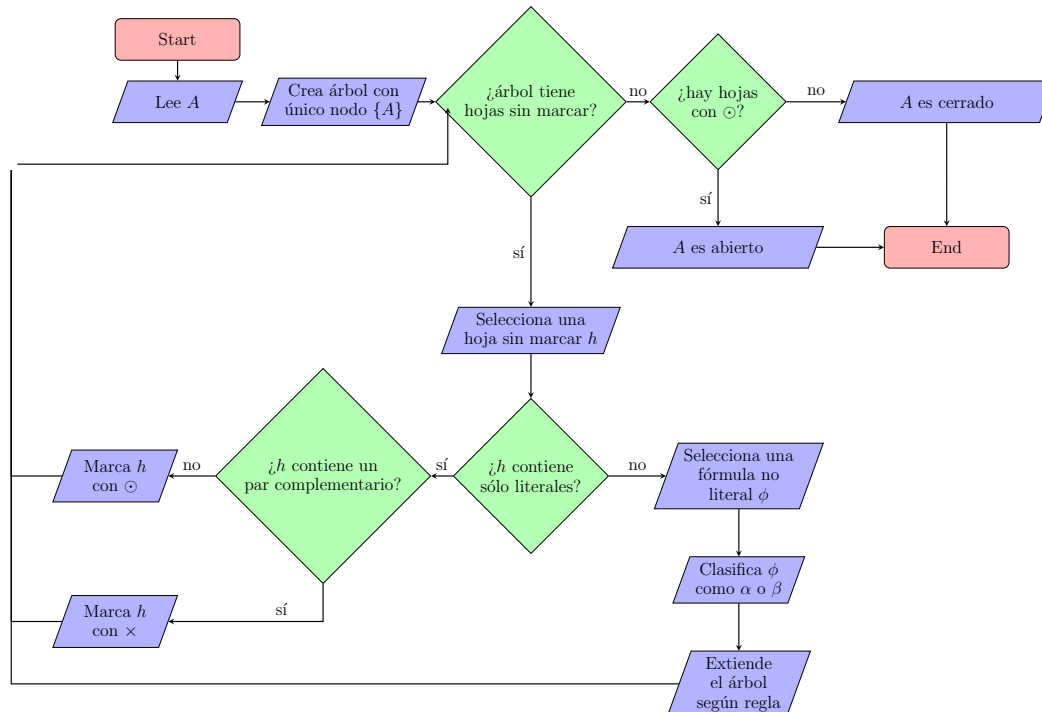
6. Compruebe en la ventana del navegador que los archivos en cuestión han sido añadidos al repositorio.

## Parte 2

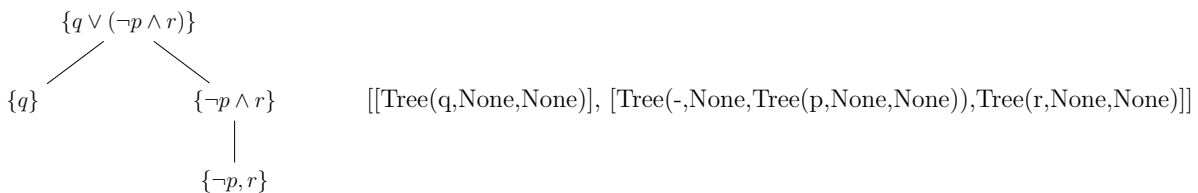
Ahora debe reunirse con su compañero de proyecto, revisar el siguiente texto y realizar los ejercicios 1 a 5. Comencemos por recordar el algoritmo de construcción de tableaux:

---

<sup>2</sup>Observe que el “>” no debe ser ingresado, sino que representa el cuadro de texto de la ventana del terminal en donde va a introducirse el comando.



Ahora, es importante observar que desarrollaremos el algoritmo de construcción de tableaux sobre la lista de hojas del tableaux. Si cada hoja se ve en sí misma como una lista de fórmulas, entonces la estructura de datos sobre la cual se desarrolla el algoritmo es una lista de *listas* de fórmulas. Por ejemplo, la estructura de la derecha corresponde al tableaux de la izquierda (observe que en la estructura de la derecha sólo se consideran las hojas del tableaux de la izquierda):



Ahora bien, dada una `listaHojas`, implementaremos la marcación de hojas de la siguiente manera. Antes que nada, necesitamos una lista vacía llamada `listaInterpsVerdaderas`.

Observe que en el archivo `tableaux.py` ya está inicializada una variable global llamada `listaInterpsVerdaderas`.

El propósito de esta lista es que, cada vez que debemos marcar una hoja con  $\odot$ , lo que hacemos es removerla de `listaHojas` e incluirla en `listaInterpsVerdaderas`. Adicionalmente, marcar una hoja con  $\times$  es simplemente removerla de `listaHojas`.

El objetivo ahora es implementar los rombos de decisión que aparecen en el algoritmo de construcción de tableaux. Gracias a la representación que hemos escogido, sólo es necesario implementar decisiones complejas para las decisiones “¿h contiene sólo literales?”

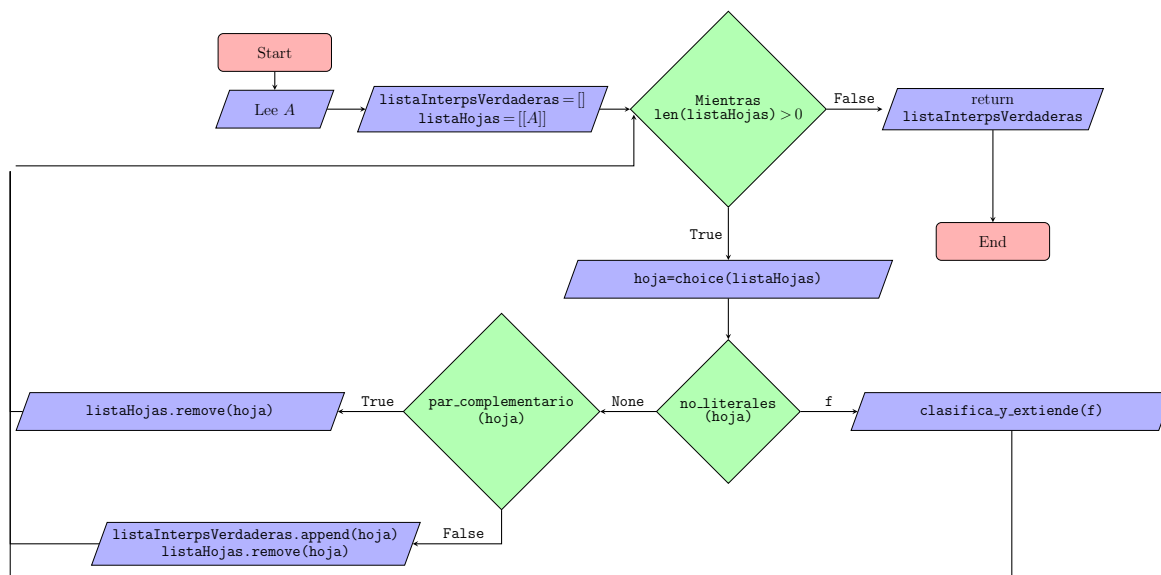
y “¿ $h$  contiene un par complementario?”, las cuales empaquetaremos en las funciones `par_complementario` y `no_literales` (ver los ejercicios 1 a 3 para la implementación de estas funciones).

Revise el archivo `tableaux.py` y encontrará la plantilla de dichas funciones.

Ahora la parte carnuda del algoritmo: clasificar una fórmula como  $\alpha$  o  $\beta$ , y extender el tableaux de acuerdo a la regla respectiva. Este procedimiento se empaqueta en la función `clasifica_y_extiende` (ver ejercicios 4 y 5).

Las reglas de formación de tableaux están en las diapositivas de la sesión 15, pág. 16 para las reglas  $\alpha$  y pág. 18 para las reglas  $\beta$ .

El diagrama de flujo del algoritmo Tableaux queda así:



Ahora debe dividirse la programación de estas funciones con su compañero:

Función	Responsable
<code>par_complementario</code>	_____
<code>no_literales</code>	_____
<code>clasifica_y_extiende</code>	_____
<code>tableaux</code>	_____

## Parte 3

Es momento de remangarse, destapar la lata de redbull, y ponerse a programar. Pero antes es importante realizar lo siguiente. Observe que cada estudiante tiene un repositorio **Tableaux**. Es esencial que entre los dos escojan cuál de los dos repositorios van a usar y añadan al otro participante como colaborador en el repositorio seleccionado.<sup>3</sup>

En el repositorio **Tableaux** seleccionado están los archivos `main.py` y `tableaux.py`. El único archivo que deben modificar es el segundo. Si cada estudiante modifica sólo la o las rutinas que seleccionó en la Parte 2, todo debe salir a la perfección. Presentaremos a continuación los ejercicios relacionados con la programación de las rutinas `par_complementario`, `no_literales`, y `clasifica_y_extiende`.

### `par_complementario`

Esta rutina corresponde a la decisión “¿ $h$  contiene un par complementario?”, donde  $h$  es una hoja que contiene sólo literales. Observe que, en su representación ‘pythonesca’,  $h$  es una lista de *literales*.

**EJERCICIO 1:** Implemente en Python la función `par_complementario`, la cual tiene como input una lista de literales  $h$ , y determina si en  $h$  hay por lo menos un par complementario (y finaliza en el momento en que encuentra el primero). Verifique su función con las listas  $[p, q]$ ,  $[p, q, \neg p]$  y  $[p, q, \neg p, \neg q]$ . En la primera no hay un par complementario, en la segunda y la tercera sí. La ejecución sobre la última lista debe detenerse tan pronto verifique que está el par  $p, \neg p$ .

### `no_literales`

Esta rutina corresponde a la decisión “¿ $h$  contiene sólo literales?”. En este caso,  $h$  es una hoja cualquiera, que contiene fórmulas. Observe que, en su representación ‘pythonesca’,  $h$  es una lista de *fórmulas*.

**EJERCICIO 2:** Implemente en Python la función `es_literal`, la cual tiene como input una fórmula  $A$  (en forma de árbol) y determina si  $A$  es un literal. Verifique su función con las fórmulas  $p$ ,  $\neg p$ ,  $\neg\neg p$ ,  $\neg(p \wedge q)$ . Las dos primeras son literales, las dos siguientes no.

**EJERCICIO 3:** Implemente en Python la función `no_literales`, la cual tiene como input una lista de fórmulas  $h$  y determina si en  $h$  hay alguna fórmula que no es un literal (y finalice en el momento en que encuentre la primera). Verifique su función con las listas  $[p, q]$ ,  $[\neg p, q]$ ,  $[p, \neg\neg q]$ ,  $[\neg\neg p, \neg(p \wedge q)]$ . En las dos primeras todas las fórmulas son literales, en las dos siguientes hay fórmulas que no son literales. La ejecución sobre la última lista debe detenerse tan pronto verifique que  $\neg\neg p$  no es un literal.

<sup>3</sup>Ir a “Settings” y luego presionar en “Invite a collaborator”.

### clasifica\_y\_extiende

Esta función tiene como input una fórmula como árbol. Aquí hay que hacer una serie de decisiones que comienza con un `if`, seguido de muchos `elifs`; una decisión para cada tipo de fórmula. Cada decisión podrá tener más `ifs` y `elifs` anidados. Por ejemplo, determinar si un `A=Tree(label,left,right)` es de categoría  $1\alpha$  requiere considerar el código de la izquierda; observe que aumentar el código para determinar si `A` es  $3\alpha$  no es muy difícil (ver código de la derecha):

```
SI A.LABEL == '-':
    SI A.RIGHT.LABEL == '-':
        RETORNAR '1ALFA'
    SI A.RIGHT.LABEL == 'O':
        RETORNAR '3ALFA'
    SI A.RIGHT.LABEL == '1':
        RETORNAR '1ALFA'
    SI A.RIGHT.LABEL == '2':
        RETORNAR '2ALFA'
    SI A.RIGHT.LABEL == '3':
        RETORNAR '3ALFA'
    SI A.RIGHT.LABEL == '4':
        RETORNAR '4ALFA'
    SI A.RIGHT.LABEL == '5':
        RETORNAR '5ALFA'
    SI A.RIGHT.LABEL == '6':
        RETORNAR '6ALFA'
    SI A.RIGHT.LABEL == '7':
        RETORNAR '7ALFA'
    SI A.RIGHT.LABEL == '8':
        RETORNAR '8ALFA'
    SI A.RIGHT.LABEL == '9':
        RETORNAR '9ALFA'
    SI A.RIGHT.LABEL == '10':
        RETORNAR '10ALFA'
    SI A.RIGHT.LABEL == '11':
        RETORNAR '11ALFA'
    SI A.RIGHT.LABEL == '12':
        RETORNAR '12ALFA'
    SI A.RIGHT.LABEL == '13':
        RETORNAR '13ALFA'
    SI A.RIGHT.LABEL == '14':
        RETORNAR '14ALFA'
    SI A.RIGHT.LABEL == '15':
        RETORNAR '15ALFA'
    SI A.RIGHT.LABEL == '16':
        RETORNAR '16ALFA'
    SI A.RIGHT.LABEL == '17':
        RETORNAR '17ALFA'
    SI A.RIGHT.LABEL == '18':
        RETORNAR '18ALFA'
    SI A.RIGHT.LABEL == '19':
        RETORNAR '19ALFA'
    SI A.RIGHT.LABEL == '20':
        RETORNAR '20ALFA'
    SI A.RIGHT.LABEL == '21':
        RETORNAR '21ALFA'
    SI A.RIGHT.LABEL == '22':
        RETORNAR '22ALFA'
    SI A.RIGHT.LABEL == '23':
        RETORNAR '23ALFA'
    SI A.RIGHT.LABEL == '24':
        RETORNAR '24ALFA'
    SI A.RIGHT.LABEL == '25':
        RETORNAR '25ALFA'
    SI A.RIGHT.LABEL == '26':
        RETORNAR '26ALFA'
    SI A.RIGHT.LABEL == '27':
        RETORNAR '27ALFA'
    SI A.RIGHT.LABEL == '28':
        RETORNAR '28ALFA'
    SI A.RIGHT.LABEL == '29':
        RETORNAR '29ALFA'
    SI A.RIGHT.LABEL == '30':
        RETORNAR '30ALFA'
    SI A.RIGHT.LABEL == '31':
        RETORNAR '31ALFA'
    SI A.RIGHT.LABEL == '32':
        RETORNAR '32ALFA'
    SI A.RIGHT.LABEL == '33':
        RETORNAR '33ALFA'
    SI A.RIGHT.LABEL == '34':
        RETORNAR '34ALFA'
    SI A.RIGHT.LABEL == '35':
        RETORNAR '35ALFA'
    SI A.RIGHT.LABEL == '36':
        RETORNAR '36ALFA'
    SI A.RIGHT.LABEL == '37':
        RETORNAR '37ALFA'
    SI A.RIGHT.LABEL == '38':
        RETORNAR '38ALFA'
    SI A.RIGHT.LABEL == '39':
        RETORNAR '39ALFA'
    SI A.RIGHT.LABEL == '40':
        RETORNAR '40ALFA'
    SI A.RIGHT.LABEL == '41':
        RETORNAR '41ALFA'
    SI A.RIGHT.LABEL == '42':
        RETORNAR '42ALFA'
    SI A.RIGHT.LABEL == '43':
        RETORNAR '43ALFA'
    SI A.RIGHT.LABEL == '44':
        RETORNAR '44ALFA'
    SI A.RIGHT.LABEL == '45':
        RETORNAR '45ALFA'
    SI A.RIGHT.LABEL == '46':
        RETORNAR '46ALFA'
    SI A.RIGHT.LABEL == '47':
        RETORNAR '47ALFA'
    SI A.RIGHT.LABEL == '48':
        RETORNAR '48ALFA'
    SI A.RIGHT.LABEL == '49':
        RETORNAR '49ALFA'
    SI A.RIGHT.LABEL == '50':
        RETORNAR '50ALFA'
    SI A.RIGHT.LABEL == '51':
        RETORNAR '51ALFA'
    SI A.RIGHT.LABEL == '52':
        RETORNAR '52ALFA'
    SI A.RIGHT.LABEL == '53':
        RETORNAR '53ALFA'
    SI A.RIGHT.LABEL == '54':
        RETORNAR '54ALFA'
    SI A.RIGHT.LABEL == '55':
        RETORNAR '55ALFA'
    SI A.RIGHT.LABEL == '56':
        RETORNAR '56ALFA'
    SI A.RIGHT.LABEL == '57':
        RETORNAR '57ALFA'
    SI A.RIGHT.LABEL == '58':
        RETORNAR '58ALFA'
    SI A.RIGHT.LABEL == '59':
        RETORNAR '59ALFA'
    SI A.RIGHT.LABEL == '60':
        RETORNAR '60ALFA'
    SI A.RIGHT.LABEL == '61':
        RETORNAR '61ALFA'
    SI A.RIGHT.LABEL == '62':
        RETORNAR '62ALFA'
    SI A.RIGHT.LABEL == '63':
        RETORNAR '63ALFA'
    SI A.RIGHT.LABEL == '64':
        RETORNAR '64ALFA'
    SI A.RIGHT.LABEL == '65':
        RETORNAR '65ALFA'
    SI A.RIGHT.LABEL == '66':
        RETORNAR '66ALFA'
    SI A.RIGHT.LABEL == '67':
        RETORNAR '67ALFA'
    SI A.RIGHT.LABEL == '68':
        RETORNAR '68ALFA'
    SI A.RIGHT.LABEL == '69':
        RETORNAR '69ALFA'
    SI A.RIGHT.LABEL == '70':
        RETORNAR '70ALFA'
    SI A.RIGHT.LABEL == '71':
        RETORNAR '71ALFA'
    SI A.RIGHT.LABEL == '72':
        RETORNAR '72ALFA'
    SI A.RIGHT.LABEL == '73':
        RETORNAR '73ALFA'
    SI A.RIGHT.LABEL == '74':
        RETORNAR '74ALFA'
    SI A.RIGHT.LABEL == '75':
        RETORNAR '75ALFA'
    SI A.RIGHT.LABEL == '76':
        RETORNAR '76ALFA'
    SI A.RIGHT.LABEL == '77':
        RETORNAR '77ALFA'
    SI A.RIGHT.LABEL == '78':
        RETORNAR '78ALFA'
    SI A.RIGHT.LABEL == '79':
        RETORNAR '79ALFA'
    SI A.RIGHT.LABEL == '80':
        RETORNAR '80ALFA'
    SI A.RIGHT.LABEL == '81':
        RETORNAR '81ALFA'
    SI A.RIGHT.LABEL == '82':
        RETORNAR '82ALFA'
    SI A.RIGHT.LABEL == '83':
        RETORNAR '83ALFA'
    SI A.RIGHT.LABEL == '84':
        RETORNAR '84ALFA'
    SI A.RIGHT.LABEL == '85':
        RETORNAR '85ALFA'
    SI A.RIGHT.LABEL == '86':
        RETORNAR '86ALFA'
    SI A.RIGHT.LABEL == '87':
        RETORNAR '87ALFA'
    SI A.RIGHT.LABEL == '88':
        RETORNAR '88ALFA'
    SI A.RIGHT.LABEL == '89':
        RETORNAR '89ALFA'
    SI A.RIGHT.LABEL == '90':
        RETORNAR '90ALFA'
    SI A.RIGHT.LABEL == '91':
        RETORNAR '91ALFA'
    SI A.RIGHT.LABEL == '92':
        RETORNAR '92ALFA'
    SI A.RIGHT.LABEL == '93':
        RETORNAR '93ALFA'
    SI A.RIGHT.LABEL == '94':
        RETORNAR '94ALFA'
    SI A.RIGHT.LABEL == '95':
        RETORNAR '95ALFA'
    SI A.RIGHT.LABEL == '96':
        RETORNAR '96ALFA'
    SI A.RIGHT.LABEL == '97':
        RETORNAR '97ALFA'
    SI A.RIGHT.LABEL == '98':
        RETORNAR '98ALFA'
    SI A.RIGHT.LABEL == '99':
        RETORNAR '99ALFA'
    SI A.RIGHT.LABEL == '100':
        RETORNAR '100ALFA'
```

EJERCICIO 4: Implemente en Python una función que, dada una fórmula que no es un literal  $A$ , la clasifique como  $1\alpha$ ,  $2\alpha$ ,  $3\alpha$  o  $4\alpha$ , o como  $1\beta$ ,  $2\beta$  o  $3\beta$ . Verifique su función con las fórmulas:

$\neg\neg(\neg(p \vee q) \wedge \neg(r \rightarrow s))$	$1\alpha$	$\neg(p \wedge (r \rightarrow s))$	$1\beta$
$\neg(p \vee q) \wedge \neg(r \rightarrow s)$	$2\alpha$	$\neg(p \wedge q) \vee (r \rightarrow s)$	$2\beta$
$\neg(\neg(r \rightarrow s) \vee q)$	$3\alpha$	$r \rightarrow (s \vee q)$	$3\beta$
$\neg(r \rightarrow \neg(p \vee q))$	$4\alpha$		

EJERCICIO 5: Implemente en Python la función `clasifica_y_extiende`, que tiene como input una hoja  $h$  y una fórmula  $f$  en  $h$  que no es un literal. Esta función debe extender la variable global `listaHojas` de acuerdo a la regla apropiada. Básicamente lo que debe hacer es borrar a  $f$  de  $h$  y, luego, o bien aumentar esta hoja o bien crear dos hojas con las fórmulas requeridas. Verifique su función con los siguientes valores de  $f$  y `listaHojas`:

$f$	<code>listaHojas</code> antes	<code>listaHojas</code> después
$p$	<code>[[p], [q]]</code>	<code>[[p], [q]]</code>
$q \wedge p$	<code>[[p], [q \wedge p]]</code>	<code>[[p], [q, p]]</code>
$\neg(p \vee (r \wedge s))$	<code>[[p, q], [\neg(p \vee (r \wedge s))]]</code>	<code>[[p, q], [\neg p, \neg(r \wedge s)]]</code>

## Parte 4

Finalmente, cuando haya terminado de implementar sus rutinas, utilice los comandos ADD, COMMIT, PUSH aprendidos en la Parte 1 para actualizar el repositorio con la copia local de su archivo `tableaux.py`. Luego, mediante la ventana del navegador, verifique que el código de todas las rutinas haya quedado incorporado en el archivo `tableaux.py` del repositorio en la nube.

Es hora de reunirse de nuevo y trabajar en un sólo computador. Allí deben actualizar el repositorio local mediante la instrucción en el terminal:

```
> git pull
```

Corran el archivo `main.py` para verificar que el código funciona sin errores. Ahora deben resolver los siguientes ejercicios.

EJERCICIO 6: Verifique que su código procesa las siguientes listas de listas de fórmulas adecuadamente:

$[[p], [q]]$	0 iteraciones y devuelve $[[p], [q]]$
$[[p], [q \wedge p]]$	1 iteración y devuelve $[[p], [q, p]]$
$[[p, q], [\neg(p \vee (r \wedge s))]]$	2 iteraciones y devuelve $[[p, q], [\neg p, \neg r], [\neg p, \neg s]]$
$[[p], [\neg(\neg(r \rightarrow s) \vee q)]]$	3 iteraciones y devuelve $[[p], [\neg r, \neg q], [s, \neg q]]$

EJERCICIO 7: Verifique el funcionamiento de la rutina de **Tableaux** con las fórmulas  $(p \vee q) \wedge (\neg p \wedge \neg q)$ ,  $\neg(p \wedge \neg p)$ ,  $\neg\neg(\neg(p \vee q) \wedge \neg(r \rightarrow s))$ . El tableaux para la primera es cerrado, mientras que los de las dos últimas son abiertos.

Para terminar, realicen el quiz virtual en e-aulas “Quiz Tableaux”.