

Estructuras Discretas 2021-2

Práctica 5, 3a parte: Tablas de verdad

Facultad de Ciencias, UNAM

Fecha de entrega: 13 de agosto del 2021

Resuelva los siguientes ejercicios, la práctica es en **parejas**. En el archivo `formulas.py` se encuentra la firma correspondiente para cada función y donde tiene que completar con código. Después de haber resuelto todo, puede ejecutar dentro de la carpeta el comando `python3 -m unittest -v test_formulas.py` para asegurarse que la salida de cada función es correcta, es importante mencionar que esto no asegura que esté bien hecho, por ejemplo, que no haya usado funciones ya definidas en **Python** que resuelvan directamente el problema pero podrá darse cuenta si alguna función es incorrecta.

Esta práctica utiliza el código de las últimas dos prácticas.

Para la entrega, siga las especificaciones publicadas en **Classroom**.

Recomendación: Cuando se usa el símbolo `\` en una cadena de python, este puede detectarlo como que se está cortando la cadena para continuarla en la siguiente línea, por lo que puede arrojar un error. Para que no suceda esto, utilice doble diagonal `\\`.

1. Defina la función `evalua_sub`. Recibe como entrada una lista de booleanos (asignación de verdad) y devuelve una lista de booleanos; las entradas de esta lista de booleanos corresponden con las posiciones de la lista de subfórmulas que genera la función `aplana`. La finalidad de esta función es generar los renglones de la tabla de verdad de esta fórmula. Sólo la primera ocurrencia está evaluada en la lista.

```
# Creación de fórmulas
>>> f1 = Formula(1)
>>> f2 = Formula(2)
>>> f4 = Formula(4)
# Conjunción de f1, f2 = f3
>>> f3 = Formula(f1, 'C', f2)
# evalua_sub de f3
>>> f3.evalua_sub([0,1])
{x1: 0, x2: 1, (x1 ∧ x2): 0}
>>> f3.evalua_sub([1,1])
{x1: 1, x2: 1, (x1 ∧ x2): 1}
# Disyunción de f3, f4 = f5
>>> f5 = Formula(f3, 'D', f4)
# evalua_sub de f5
>>> f5.evalua_sub([1,1,0])
{x1: 1, x2: 1, (x1 ∧ x2): 1, x4: 0, ((x1 ∧ x2) ∨ x4): 1}
```

2. Defina la función `renglones_verdad` que devuelve una lista con los renglones de la tabla de verdad de la fórmula. Por diseño, los valores de las variables sólo ocurren en las primeras columnas de la tabla de verdad.

```
# Creación de fórmulas
>>> f1 = Formula(1)
>>> f2 = Formula(2)
# Conjunción de f1, f2 = f3
>>> f3 = Formula(f1, 'C', f2)
# renglones_verdad de f3
>>> f3.renglones_verdad()
[[0, 0, x1, x2, (x1 ∧ x2)], [0, 1, x1, x2, (x1 ∧ x2)], [1, 0, x1, x2, (x1 ∧ x2)], [1, 1,
x1, x2, (x1 ∧ x2)]]
# Para mayor claridad, la lista se puede pensar así (es la misma lista):
[[0, 0, x1, x2, (x1 ∧ x2)],
 [0, 1, x1, x2, (x1 ∧ x2)],
 [1, 0, x1, x2, (x1 ∧ x2)],
 [1, 1, x1, x2, (x1 ∧ x2)]]
```

3. Implemente de manera recursiva la función `tex_tabla` que devuelva la fórmula en LaTeX con separadores de tabla. Esta función se utilizará para construir el encabezado de la tabla de verdad.

```

>>> x1 = Formula(1)
>>> x2 = Formula(2)
>>> x3 = Formula(3)
>>> f1 = Formula(x1, 'C', x2)
>>> f2 = Formula(x3, 'N')
>>> f3 = Formula(f1, 'D', f2)

>>> f1.tex_tabla()
(x_{1} & \land & x_{2})
>>> f2.tex_tabla()
( & \lnot & x_{3})
>>> f3.tex_tabla()
((x_{1} & \land & x_{2}) & \lor & ( & \lnot & x_{3}))

```

4. Implemente la función `_cabecera_tabla` que devuelva una cadena en LaTeX que represente la cabecera de la tabla de verdad de la fórmula. Debe incluir las variables y la fórmula completa. No es necesario usar recursión

```

>>> x1 = Formula(1)
>>> x2 = Formula(2)
>>> x3 = Formula(3)
>>> f1 = Formula(x1, 'C', x2)
>>> f2 = Formula(x3, 'N')
>>> f3 = Formula(f1, 'D', f2)

>>> f1._cabecera_tabla()
x_{1} & x_{2} & (x_{1} & \land & x_{2}) \\
>>> f2._cabecera_tabla()
x_{3} & ( & \lnot & x_{3}) \\
>>> f3._cabecera_tabla()
x_{1} & x_{2} & x_{3} & ((x_{1} & \land & x_{2}) & \lor & ( & \lnot & x_{3})) \\

```

5. Implemente la función `_renglon_verdad` que dada una asignación de valores de verdad a las variables, devuelva el renglón de la tabla de verdad en LaTeX. No es necesario usar recursión.

Hint: Utilizar `aplana_sin_variables` y `evalua_sub` para obtener las subfórmulas y su evaluación dada la asignación de verdad.

```

>>> x1 = Formula(1)
>>> x2 = Formula(2)
>>> x3 = Formula(3)
>>> f1 = Formula(x1, 'C', x2)
>>> f2 = Formula(x3, 'N')
>>> f3 = Formula(f1, 'D', f2)

>>> f1._renglon_verdad((0,0))
0 & 0 & & \mathbf{0} & \\
>>> f1._renglon_verdad((1,0))
1 & 0 & & \mathbf{0} & \\
>>> f3._renglon_verdad((1,0,1))
1 & 0 & 1 & & 0 & & \mathbf{0} & & 0 & \\

```

6. Implemente la función `tabla_verdad` que devuelva la tabla de verdad de la fórmula en LaTeX. No es necesario usar recursión.

```

>>> x1 = Formula(1)
>>> x2 = Formula(2)
>>> x3 = Formula(3)
>>> f1 = Formula(x1, 'C', x2)
>>> f2 = Formula(x3, 'N')
>>> f3 = Formula(f1, 'D', f2)

>>> f1.tabla_verdad()
\begin{adjustbox}{max width=\textwidth,array=cc|ccc} \\
x_{1} & x_{2} & (x_{1} & \land & x_{2}) & \\
\hline
1 & 1 & & \mathbf{1} & \\
1 & 0 & & \mathbf{0} & \\
0 & 1 & & \mathbf{0} & \\
0 & 0 & & \mathbf{0} & \\
\end{adjustbox}

```

```

>>> f3.tabla_verdad()
\begin{adjustbox}{max width=\textwidth,array=ccc|cccc@{}cc} \\
x_{1} & x_{2} & x_{3} & ((x_{1} & \land & x_{2}) & \lor & ( & \neg & x_{3})) \\
\hline
1 & 1 & 1 & & \mathbf{1} & & 0 & \\
1 & 1 & 0 & & \mathbf{1} & & 1 & \\
1 & 0 & 1 & & \mathbf{0} & & 0 & \\
1 & 0 & 0 & & \mathbf{1} & & 1 & \\
0 & 1 & 1 & & \mathbf{0} & & 0 & \\
0 & 1 & 0 & & \mathbf{1} & & 1 & \\
0 & 0 & 1 & & \mathbf{0} & & 0 & \\
0 & 0 & 0 & & \mathbf{1} & & 1 & \\
\end{adjustbox}

```

7. Implemente la función **LaTeX** que genera un archivo con terminación **.tex** que es un documento LaTeX con la tabla de verdad de la fórmula. No es necesario usar recursión.

Hint: Utilizar las funciones **write**, **open**, **close** para manipular los archivos. También se puede utilizar la biblioteca **pathlib**.

```

>>> x1 = Formula(1)
>>> x2 = Formula(2)
>>> f1 = Formula(x1, 'C', x2)

>>> f1.LaTeX("ejemplo")
##Esto genera un archivo ejemplo.tex con el siguiente contenido
\documentclass{article}

\usepackage{adjustbox}

\begin{document}

\[
\begin{adjustbox}{max width=\textwidth,array=cc|ccc} \\
x_{1} & x_{2} & (x_{1} & \land & x_{2}) \\
\hline
1 & 1 & & \mathbf{1} & \\
1 & 0 & & \mathbf{0} & \\
0 & 1 & & \mathbf{0} & \\
0 & 0 & & \mathbf{0} & \\
\end{adjustbox}
\]

\end{document}

```