Parsetree.py

The code mainly use a structure similar to a binary arithmetic expression tree, but replacing with logical operators. Stacks were used to convert the infix logical expression to postfix and store in the tree.

Operator nodes and operands nodes are different. Parsing is done from scratch.

Input format

Propositional arguments: Each variable/operator has to be separated by white spaces before and after.

AND – '&'
OR – '|'
NOT – !
IMPLIES – >

> Example: (P1 and not P2) will be
> (P1 & ! P2)
> (P1 & ! P2) will produce error since bracket is not separated.

- Truth assignments: Must be in the form of 'Pn = T,' T or True , F or False can be used . Usually truth assignments are taken from user input.
- Uppercases are handled, namely, p1 and P1 are the same variable. T an t both stands for true.
-

Dependancies

Python >= 3.8, sys

Usage

The code comes with different modes.

Parsetree.py (-h | --help)
Parsetree.py (-e | --example)
Parsetree.py -i
Parsetree.py -f P_file [S_file]
Parsetree.py -n1 string_P string_S
Parsetree.py -n2 string_P

Options:
-h --help Show this help message
-e --example Show examples in the manual
-i Take P and S from guided user input
-f Take P ans S from txt files
-n1 Take P and S from command lines
-n2 Take P to generate truth table

> Example
> python3 Parsetree.py -e
> ...
> Starting Part 1 example

> The example expression used here is:
> ((P1 & p2) | (p3 & T)) | ((P1 & ! P3) & P2)
> P1 has no corresponding truth assignment.
> Enter T to assign True or F to assign False to it.
> Or A to abort the current session.
>
> > T
> > ...
> > The truth assignment used was: {'P2': True, 'P1': True, 'P3': True}.
> > The evaluation of the expression is True
> > Part 1 example done.
> > Press Enter to Continue...

More

> python3 Parsetree.py -f [file1.txt] file2.txt

Each line in file2 is treated as a logical expression, take the corresponding line position in file 1 as the truth assignment, and try to evaluate the result. If file1 ist not provided, then a truth table is generated for each logical expression in file2.

Files have to be .txt files, each line correspond to one truth assignment or one logical expression

> python3 Parsetree.py -i

A guided mode that receive all the argument from input().

> Please enter the proposition string you wish to process. Each variable and operator has to be separated by the white space.
>
> > P1 & P2
> > The proposition entered is: P1 & P2
> > Would you like to
> > 1-generate a truth table or 2-keep entering truth assignment?
> > Enter 1 or 2
> >
> > > 1
> > > P1 P2 P1 & P2
> > > True True True
> > > True False False
> > > False True False
> > > False False False
> >
> > > The P1 & P2 is CONTINGENCY
> >
> > > Session completed.

The -n1 and -n2 mode take arguments from command lline.