

COMP 5361 Discrete Structures and Formal Languages

Programming Assignment-2 Report

Submitted To: Prof. Gösta Grahne

Yash Jayeshkumar Pandya (40119272)

Problem statement:

1. Write a Python program that takes as input a truth assignment A for propositional variables P = {P1, P2, . . . , P n} and a propositional sentence S involving these variables, and produces output True or False, depending on whether the given assignment A satisfies the sentence S or not. To show that your program works, run your program on the following input

((P1
$$\wedge$$
 P2) \vee (P3 \wedge True)) \vee ((¬P1 \wedge ¬P3) \wedge P2).

Note that your program should work for any $n \ge 1$, and other inputs also will be given as tests.

2. Write a Python program that takes input P and S as in Question 1, generates the truth table for S, and outputs Tautology, Contingency, or Contradiction, depending on which category S falls into. To show that your program works, run your program on the following inputs

(a)
$$(\neg P1 \land (P1 \lor P2)) \rightarrow P2$$

(b) P2
$$\land$$
 (P1 $\rightarrow \neg$ P2) \land (\neg P1 $\rightarrow \neg$ P2)

(c)
$$(P1 \rightarrow (P2 \rightarrow P3)) \rightarrow ((P1 \rightarrow P2) \rightarrow P3)$$

Program Logic:

The program takes user input of either user wants to calculate value for truth assignments of the equation or truth table for the equation. Once selected, the equation is taken as an input and if the user wants to calculate the value of truth assignments then based on the number of variables, the user is asked to provide boolean value for each variable.

The program uses List and Dictionary data structure of the python language, however, list is converted as a stack implementation. Multiple stacks are used in order to generate and store postfix expression sequence and display sequence. Dictionary is used to keep the variables as a key and boolean truth assignment as a value and used for evaluating postfix expression.

- Some intermediate functions are created in-order to evaluate the Conditional, Bi-Conditional, Disjunction and other operations.
- Some constants are used in-order to map/compare static assignments.

generate_expression function takes the entered equation and calculates it's postfix expression. After generating a postfix expression, evaluate_expression function performs evaluation of the postfix expression based on the boolean value entered by the user for the variables. This function takes a boolean assignment for variables and produces the result.

Ex. ((A
$$\wedge$$
 B) => C) to postfix [A, B, \wedge , C, =>]

On the other hand, displaying the truth table option is selected by the user then <code>calculate_truth_table_and_equivalence</code> is getting called, this function generates the possibilities of the boolean values for each of the variables. If the equation contains 3 variables then possibilities are having the eight unique combinations of boolean values. After that, <code>evaluate_expression</code> function performs the evaluation of postfix expression with specific boolean values and returns to the <code>calculate_truth_table_and_equivalence</code> function and later values are getting displayed by the same function. This continues till all possible boolean combinations of values are evaluated. Later, based on the result values, the function determines either entered expression is Tautology, Contradiction or Contingency.

How to Run:

Run Program (Command: python asgn2.py)

When the program runs user is displayed with following menu in the console

```
COMP-5361 Assignment-2 Menu

1. Produce output from truth assignments
2. Display truth table and propositional equivalences
3. Exit

Select:
```

The user needs to select any one of the available options at a time, otherwise it will raise an error for invalid choice.

After selecting either of the choice 1 or 2, the user needs to give an input of a propositional logic equation. If the equation is invalid the user needs to repeat the same process again.

Please enter valid propositional logic equation : ((A \land B) => C) For choice 1, the user needs to give boolean inputs for variables in order to compute output from the truth assignments.

```
Please enter bool value for A : True
Please enter bool value for B : True
Please enter bool value for C : False
```

Then, for choice 1 below, the result will be displayed.

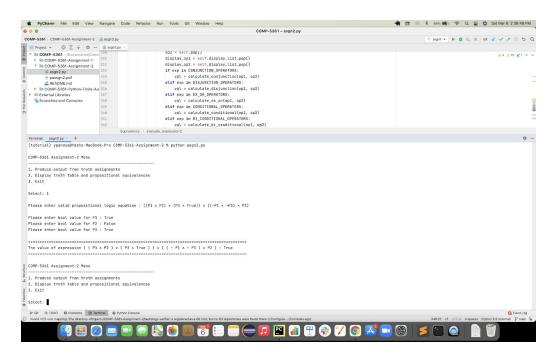
______ The value of expression ((A \wedge B) => C) : False

And, for choice 2 below, the result will be displayed.

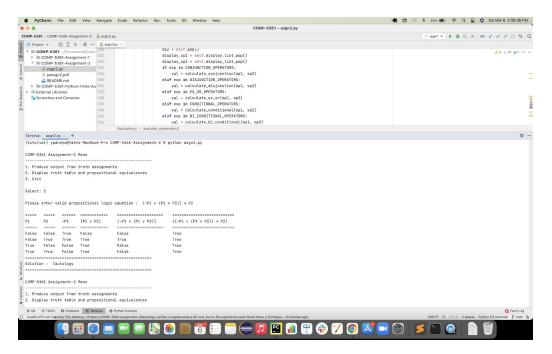
====	====	====	=======	===========
A	В	С	(A ∧ B)	$((A \land B) => C)$
====	====	====	=======	==========
False	False	False	False	True
False	False	True	False	True
False	True	False	False	True
False	True	True	False	True
True	False	False	False	True
True	False	True	False	True
True	True	False	True	False
True	True	True	True	True
Solution: Contingency				

Solution: Contingency

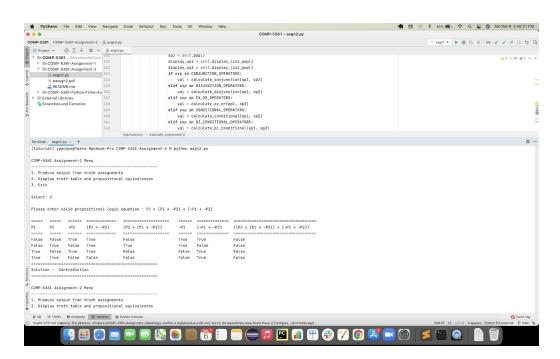
Test Cases:



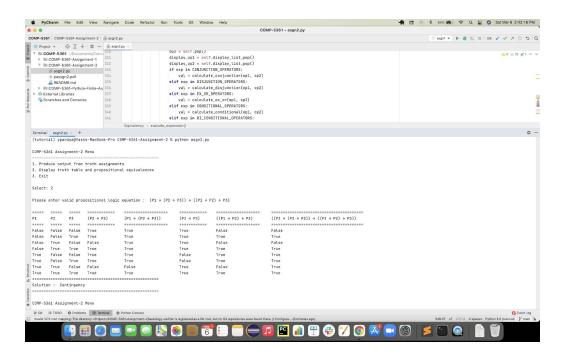
Question 1



Question 2 (a)



Question 2 (b)



Question 2 (c)

References:

- https://stackoverflow.com/questions/53526207/how-do-i-add-a-row-of-dashes-between-the-first-two-print-lines-in-python
- https://www.w3resource.com/python-exercises/challenges/1/python-challenges-1-exercise-56.ph
 p
- https://rosettacode.org/wiki/Truth_table