



CSE 15: Discrete Mathematics Laboratory 3

Spring 2019

Introduction

In this lab we continue to practice our Python programming skills by adding further functionality to our library of propositional logic tools. For this lab, there is a new version of `logic.py`, which makes creating truth tables a little more convenient as you no longer need to specify a list of the variables involved in your propositions.

Originally, to create a truth table for the expressions $p \wedge q$, and $p \vee q$, we had to do the following:

```
myTable = TruthTable(['p', 'q'], ['p and q', 'p or q'])
```

We no longer have to provide the first parameter, so now we call:

```
myTable = TruthTable(['p and q', 'p or q'])
```

and the algorithm goes through the list of propositions and figures out the variable names for us. The new version of `logic.py` is available in the folder of Lab 3, in the Files section of CatCourses.

Consistency of System Descriptions

Write a Python program that asks the user to enter a system description as a series of propositions. Determine whether or not the description is consistent and print a message to that effect. Your user experience should be along the lines of the following:

```
Enter a proposition: p and q

Would you like to enter more (Y/N): Y

Enter a proposition: p -> q

Would you like to enter more (Y/N): N

Your description is consistent.
```

Save your program in a file and upload it under the appropriate CatCourses assignment page.

Descriptive System Descriptions

Extend the functionality of your program from the last question by allowing the user to assign a meaning to all the propositional variables. If the user has entered a consistent description, your program should also outline the conditions that satisfy consistency. For example:

```
Enter a proposition: -p

Would you like to enter more (Y/N): Y

Enter a proposition: -q

Would you like to enter more (Y/N): Y

Enter a proposition: p <-> q

Would you like to enter more (Y/N): N

Your program uses propositional variables ['p', 'q']:

Enter meaning of p: John is a Knight

Enter meaning of q: James is a Knight

Your description is consistent when:

It is not the case that John is a Knight
It is not the case that James is a Knight
```

Save your program in a file and upload it under the appropriate CatCourses assignment page.

Hints: To get a list of all the variables used in the propositions the user enters, simply make a truth table for the propositions. Say your truth table is called `myTruthTable`, then the list of variables occurring in all the propositions is `myTruthTable.vars`