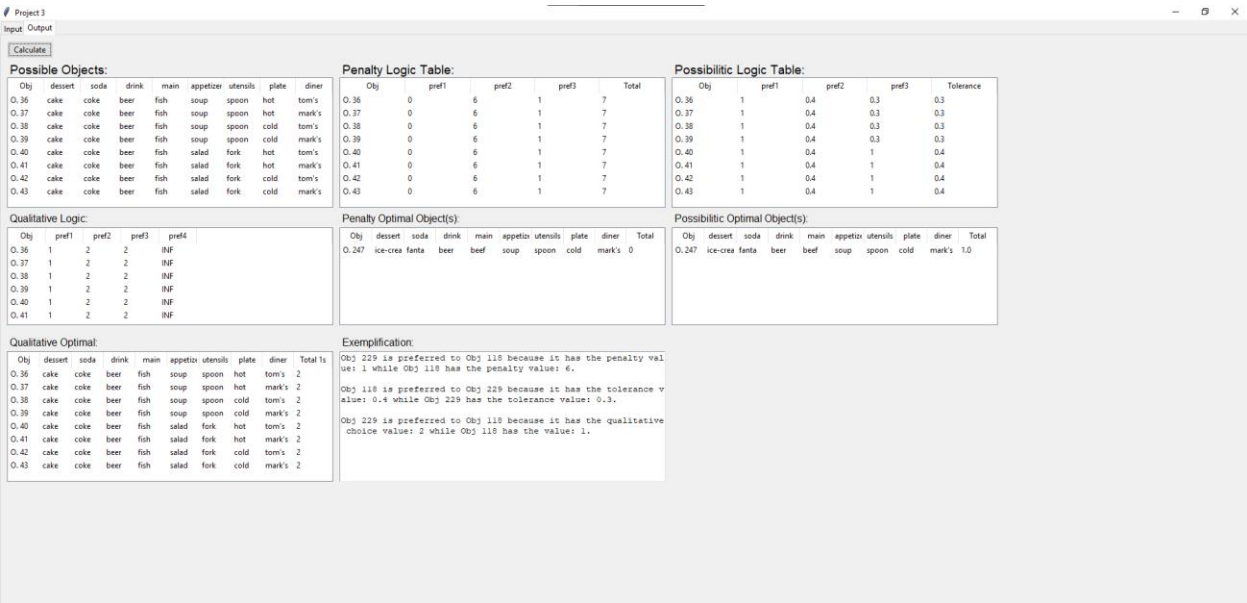
Project 3 – Logic/Preference Solver

CAP4630 Intro to AI

Liam Rigney, Ricky LeDew

Professor Xudong Liu

March 31, 2023



Project Overview

We developed a logic/preference solver which, given a test case consisting of attributes, hard constraints, and preferences (using possibilistic, penalty, and qualitative choice models), can produce objects that satisfy these constraints, optimal objects for the given preferences, and compare two randomly generated feasible objects to each other.

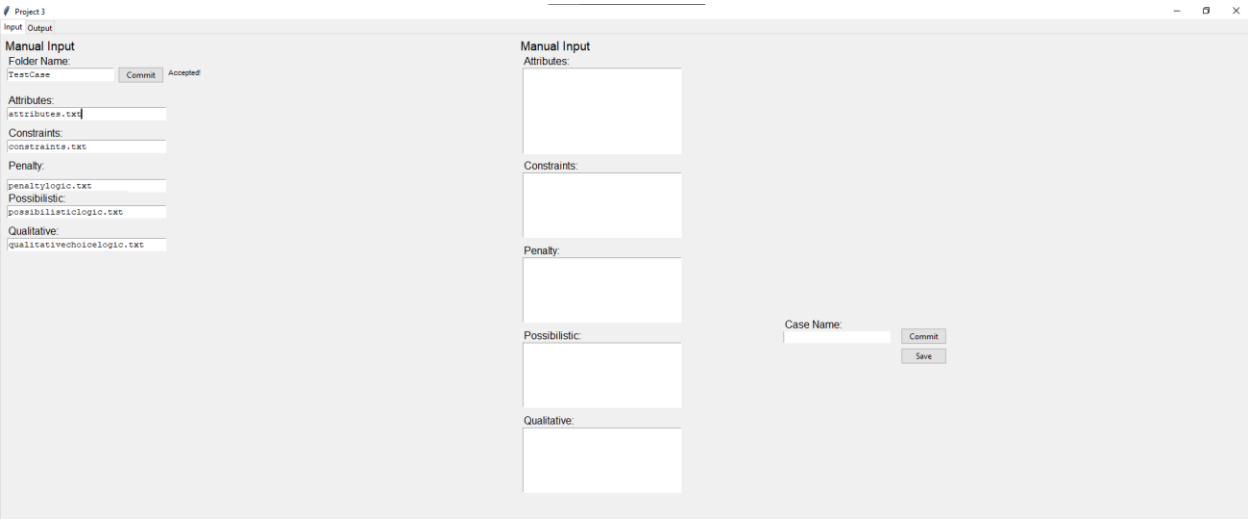
Our project allows the user to input either a predefined test case, or manually enter in attributes, hard constraints, and preferences to use.

How it Works

Inputting Attributes, Constraints, and Preferences

The main screen provides an interface which you can use to load attributes, constraints and preferences. You can do this from a folder using the inputs on the left, or manually input them through the inputs on the right. If loading from a folder, enter the folder name, and then in the other text boxes enter the names of the text files that contain the attributes, constraints, and various types of preferences. Press the “Commit” button to load these files, and then navigate to the “Output” tab and click on “Calculate”

to continue.

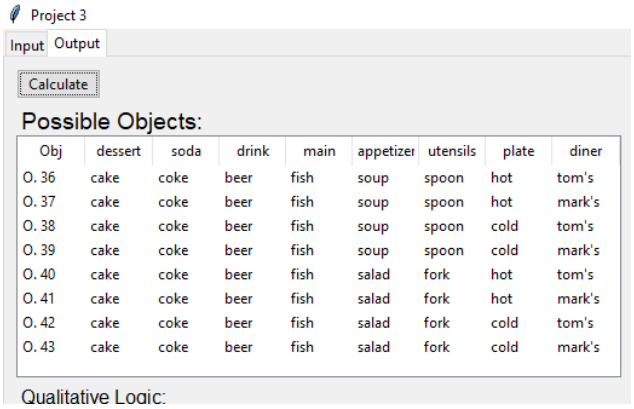


Generating Feasible Objects

Our project generates feasible objects by generating every combination of attributes, and then filtering them to find ones that satisfy the hard constraints provided by the user.

After all objects are checked, the table is populated with those that satisfy the given constraints.

When generating feasible objects, the object numberings will often not be sequential because *every object* gets its own number, not just feasible ones, however only feasible objects are shown in the table. E.g. In this example the table starts at “O. 36” because objects 0-35 do not satisfy the constraints.



Obj	dessert	soda	drink	main	appetizer	utensils	plate	diner
O. 36	cake	coke	beer	fish	soup	spoon	hot	tom's
O. 37	cake	coke	beer	fish	soup	spoon	hot	mark's
O. 38	cake	coke	beer	fish	soup	spoon	cold	tom's
O. 39	cake	coke	beer	fish	soup	spoon	cold	mark's
O. 40	cake	coke	beer	fish	salad	fork	hot	tom's
O. 41	cake	coke	beer	fish	salad	fork	hot	mark's
O. 42	cake	coke	beer	fish	salad	fork	cold	tom's
O. 43	cake	coke	beer	fish	salad	fork	cold	mark's

Penalty/Possibilistic/Qualitative Choice Logic Tables

We also have tables which display how different feasible objects perform with respect to penalty, possibilistic, and qualitative choice preferences provided by the user.

For each object, the **penalty logic** table displays the penalty value associated with each preference, as well as a total penalty value.

Penalty Logic Table:

Obj	pref1	pref2	pref3	Total
O. 36	0	6	1	7
O. 37	0	6	1	7
O. 38	0	6	1	7
O. 39	0	6	1	7
O. 40	0	6	1	7
O. 41	0	6	1	7
O. 42	0	6	1	7
O. 43	0	6	1	7

Possibilitic Logic Table:

Obj	pref1	pref2	pref3	Tolerance
O. 36	1	0.4	0.3	0.3
O. 37	1	0.4	0.3	0.3
O. 38	1	0.4	0.3	0.3
O. 39	1	0.4	0.3	0.3
O. 40	1	0.4	1	0.4
O. 41	1	0.4	1	0.4
O. 42	1	0.4	1	0.4
O. 43	1	0.4	1	0.4

The **possibilistic logic** table displays the tolerance values for each object, according to each preference. For the rightmost column, the minimum tolerance value is chosen to represent that object.

The **qualitative choice logic** table displays the satisfaction degrees for each object, according to each preference. INF is used in places where '∞' should appear.

Qualitative Logic:

Obj	pref1	pref2	pref3	pref4
O. 36	1	2	2	INF
O. 37	1	2	2	INF
O. 38	1	2	2	INF
O. 39	1	2	2	INF
O. 40	1	2	2	INF
O. 41	1	2	2	INF

Generating Optimal Objects

Our program also generates sets of optimal objects according to the provided preferences. To do this, it takes the rows of the penalty, possibilistic, and qualitative choice tables, and finds an object which has the minimum penalty, maximum tolerance, or best satisfaction degrees.

Penalty Optimal Object(s):

Obj	dessert	soda	drink	main	appetize	utensils	plate	diner	Total
O. 247	ice-crea	fanta	beer	beef	soup	spoon	cold	mark's	0

Possibilitic Optimal Object(s):

Obj	dessert	soda	drink	main	appetize	utensils	plate	diner	Total
O. 247	ice-crea	fanta	beer	beef	soup	spoon	cold	mark's	1.0

For qualitative choice, since there isn't a "total", we use the number of '1's as a measure of how preferred the object is overall. E.g., an object with 2 preferences with satisfaction degree 1 would beat an object with only one preference with an S.D. of 1. There may be many objects that share the same # of 1s, and so they are all displayed in this table.

Qualitative Optimal:

Obj	dessert	soda	drink	main	appetize	utensils	plate	diner	Total 1s
O. 36	cake	coke	beer	fish	soup	spoon	hot	tom's	2
O. 37	cake	coke	beer	fish	soup	spoon	hot	mark's	2
O. 38	cake	coke	beer	fish	soup	spoon	cold	tom's	2
O. 39	cake	coke	beer	fish	soup	spoon	cold	mark's	2
O. 40	cake	coke	beer	fish	salad	fork	hot	tom's	2
O. 41	cake	coke	beer	fish	salad	fork	hot	mark's	2
O. 42	cake	coke	beer	fish	salad	fork	cold	tom's	2
O. 43	cake	coke	beer	fish	salad	fork	cold	mark's	2

Exemplification

Our program can also generate two objects and compare them using the different preference models. We first compare using penalty logic, then possibilistic logic, and then qualitative choice logic.

Exemplification:

```
Obj 229 is preferred to Obj 118 because it has the penalty val
ue: 1 while Obj 118 has the penalty value: 6.

Obj 118 is preferred to Obj 229 because it has the tolerance v
alue: 0.4 while Obj 229 has the tolerance value: 0.3.

Obj 229 is preferred to Obj 118 because it has the qualitative
 choice value: 2 while Obj 118 has the value: 1.
```