

Program Overview

This program is designed as a knowledge-based intelligent system to solve preference problems by collecting user preferences and reasoning about them. It offers a user interface for inputting attributes, hard constraints, and preferences, which can be read from text files. The system is supposed to support two types of preference logic: penalty logic and qualitative choice logic. However, I was only able to get the penalty logic working before the deadline.

Core Components

Attributes: All attributes in the system are binary. The attributes and their values (encoded as '1' and '0') are specified in a text file.

Hard Constraints (H): Represented as propositional formulas in Conjunctive Normal Form (CNF), these constraints dictate the feasibility of various combinations of attributes.

Preference Theory (T): The system supports penalty logic and qualitative choice logic to evaluate preferences. The preferences are also described using CNF formulas.

Key Functionalities

Encoding: Converts attribute combinations into binary codes based on the order listed in the attributes file.

Feasibility Checking: Determines whether there are feasible objects with respect to the hard constraints defined.

Show the Table: Displays a table of all feasible objects, indicating their penalty or satisfaction degree values for each preference rule.

Exemplification: Generates two random feasible objects (if possible) and shows the preference between them based on the selected preference logic.

Omni-optimization: Identifies all optimal feasible objects according to the selected preference theory.

File Formats

Attributes File: Lists the binary attributes and their possible values.

Hard Constraints File: Contains CNF formulas representing the hard constraints.

Preferences File: Two separate files for penalty logic and qualitative choice logic, respectively, outlining the preference rules.

How It Works

Startup: Upon launching, the user is prompted to enter the filenames for attributes, hard constraints, and preferences. These files are parsed to set up the system's knowledge base.

User Interface: The user selects the preference logic (penalty or qualitative choice logic) and then chooses a reasoning task (encoding, feasibility checking, showing the table, exemplification, or omni-optimization).

Processing: Based on the selected task, the system processes the input data, applies the relevant logic, and outputs the results. For exemplification and omni-optimization, random feasible objects are generated and evaluated according to the preference theory in use.

Dynamic Table Generation: For tasks that require displaying information in a table format, headers are dynamically generated based on the rules specified in the preferences file, ensuring flexibility and adaptability to different sets of rules.

Running the Program

To run the program, navigate to the project root directory and execute:

➤ `python main.py`

Ensure Python 3.11 or later is installed on your system. The program does not require any external dependencies outside of the standard Python library.

Ensure all files are in the same directory. This may not be necessary, but I only tested it will all project files in the same directory.

Base Given Case

```
Command Prompt - python r x + v
(venv) C:\Users\erics\Documents\aiproject3>python main.py
Welcome to PrefAgent!
Enter Attributes File Name: |

Command Prompt - python r x + v
(venv) C:\Users\erics\Documents\aiproject3>python main.py
Welcome to PrefAgent!
Enter Attributes File Name: attributes.txt
Enter Hard Constraints File Name: constraints.txt
Choose the preference logic to use:
1. Penalty Logic
2. Qualitative Choice Logic
3. Exit
Your Choice: |
```

```
Command Prompt - python r x + v
(venv) C:\Users\erics\Documents\aiproject3>python main.py
Welcome to PrefAgent!

Enter Attributes File Name: attributes.txt
Enter Hard Constraints File Name: constraints.txt

Choose the preference logic to use:
1. Penalty Logic
2. Qualitative Choice Logic
3. Exit

Your Choice: 1

You picked Penalty Logic
Enter Preferences File Name: penaltylogic.txt

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 1
Encoded Objects:
o0: ice-cream, beer, beef
o1: ice-cream, beer, fish
o2: ice-cream, wine, beef
o3: ice-cream, wine, fish
o4: cake, beer, beef
o5: cake, beer, fish
o6: cake, wine, beef
o7: cake, wine, fish

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: |

6. Back to previous menu

Your Choice: 2
Yes, there are 6 feasible objects.

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 3
+-----+-----+-----+-----+
| encoding | fish AND wine | wine OR cake | total penalty |
+-----+-----+-----+-----+
| o0       | 10             | 6             | 16             |
| o1       | 10             | 6             | 16             |
| o4       | 10             | 0             | 10             |
| o5       | 10             | 0             | 10             |
| o6       | 10             | 0             | 10             |
| o7       | 0              | 0             | 0              |
+-----+-----+-----+-----+

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 4
Two randomly selected feasible objects are o1 and o0,
and they are equivalent.

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 4
```

```
Command Prompt - python r x + v
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 6

Choose the preference logic to use:
1. Penalty Logic
2. Qualitative Choice Logic
3. Exit

Your Choice: 2

You picked Qualitative Choice Logic
Enter Preferences File Name: qualitativechoicelogic.txt

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 1
Encoded Objects:
o0: ice-cream, beer, beef
o1: ice-cream, beer, fish
o2: ice-cream, wine, beef
o3: ice-cream, wine, fish
o4: cake, beer, beef
o5: cake, beer, fish
o6: cake, wine, beef
o7: cake, wine, fish

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: |

Yes, there are 6 feasible objects.

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 3
No penalty logic rules provided.

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 5
Traceback (most recent call last):
  File "C:\Users\erics\Documents\aiproject3\main.py", line 4, in <module>
    userInterface()
  File "C:\Users\erics\Documents\aiproject3\userInterface.py", line 83, in userInterface
    reasoningTasksMenu(attributes, encodedObjects, feasibleObjects, constraints, preferenceChoice, penaltyLogicRules=penaltyLogicRules)
  File "C:\Users\erics\Documents\aiproject3\userInterface.py", line 54, in reasoningTasksMenu
    reasoningTasksMenu(attributes, encodedObjects, feasibleObjects, constraints, preferenceChoice, qualitativeLogicRules=qualitativeLogicRules)
  File "C:\Users\erics\Documents\aiproject3\userInterface.py", line 37, in reasoningTasksMenu
```

As you can see, my qualitative logic crashes because I did not finish it in time.

My Custom Cases

```
Command Prompt - python r x + v

(venv) C:\Users\erics\Documents\aiproject3>python main.py
Welcome to PrefAgent!

Enter Attributes File Name: testattributes.txt

Enter Hard Constraints File Name: testconstraints.txt

Choose the preference logic to use:
1. Penalty Logic
2. Qualitative Choice Logic
3. Exit

Your Choice: 1

You picked Penalty Logic
Enter Preferences File Name: testpenaltylogic.txt

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: |

Command Prompt - python r x + v

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 1
Encoded Objects:
o0: beach, airBNB, surfing, weekend, car, cooking, luxury, summer
o1: beach, airBNB, surfing, weekend, car, cooking, luxury, winter
o2: beach, airBNB, surfing, weekend, car, cooking, budget, summer
o3: beach, airBNB, surfing, weekend, car, cooking, budget, winter
o4: beach, airBNB, surfing, weekend, car, allInclusive, luxury, summer
o5: beach, airBNB, surfing, weekend, car, allInclusive, luxury, winter
o6: beach, airBNB, surfing, weekend, car, allInclusive, budget, summer
o7: beach, airBNB, surfing, weekend, car, allInclusive, budget, winter
o8: beach, airBNB, surfing, weekend, plane, cooking, luxury, summer
o9: beach, airBNB, surfing, weekend, plane, cooking, luxury, winter
o10: beach, airBNB, surfing, weekend, plane, cooking, budget, summer
o11: beach, airBNB, surfing, weekend, plane, cooking, budget, winter
o12: beach, airBNB, surfing, weekend, plane, allInclusive, luxury, summer
o13: beach, airBNB, surfing, weekend, plane, allInclusive, luxury, winter
o14: beach, airBNB, surfing, weekend, plane, allInclusive, budget, summer
o15: beach, airBNB, surfing, weekend, plane, allInclusive, budget, winter
o16: beach, airBNB, surfing, week, car, cooking, luxury, summer
o17: beach, airBNB, surfing, week, car, cooking, luxury, winter
o18: beach, airBNB, surfing, week, car, cooking, budget, summer
o19: beach, airBNB, surfing, week, car, cooking, budget, winter
o20: beach, airBNB, surfing, week, car, allInclusive, luxury, summer
o21: beach, airBNB, surfing, week, car, allInclusive, luxury, winter
o22: beach, airBNB, surfing, week, car, allInclusive, budget, summer
o23: beach, airBNB, surfing, week, car, allInclusive, budget, winter
o24: beach, airBNB, surfing, week, plane, cooking, luxury, summer
o25: beach, airBNB, surfing, week, plane, cooking, luxury, winter
o26: beach, airBNB, surfing, week, plane, cooking, budget, summer
o27: beach, airBNB, surfing, week, plane, cooking, budget, winter
o28: beach, airBNB, surfing, week, plane, allInclusive, luxury, summer
o29: beach, airBNB, surfing, week, plane, allInclusive, luxury, winter
o30: beach, airBNB, surfing, week, plane, allInclusive, budget, summer
o31: beach, airBNB, surfing, week, plane, allInclusive, budget, winter
o32: beach, airBNB, snowboarding, weekend, car, cooking, luxury, summer
o33: beach, airBNB, snowboarding, weekend, car, cooking, luxury, winter
o34: beach, airBNB, snowboarding, weekend, car, cooking, budget, summer
o35: beach, airBNB, snowboarding, weekend, car, cooking, budget, winter
```

[illegible]


```
Command Prompt - python r x + v
Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 3

+-----+-----+-----+-----+-----+
| encoding | mountain AND snowboarding | beach AND surfing | luxury AND plane | total penalty |
+-----+-----+-----+-----+-----+
| o6       | 10      | 8      | 5      | 23      |
| o7       | 10      | 8      | 5      | 23      |
| o12      | 10      | 8      | 5      | 23      |
| o13      | 10      | 8      | 5      | 23      |
| o14      | 10      | 8      | 0      | 18      |
| o15      | 10      | 8      | 0      | 18      |
| o22      | 10      | 8      | 5      | 23      |
| o23      | 10      | 8      | 5      | 23      |
| o28      | 10      | 8      | 5      | 23      |
| o29      | 10      | 8      | 5      | 23      |
| o30      | 10      | 8      | 0      | 18      |
| o31      | 10      | 8      | 0      | 18      |
| o70      | 10      | 8      | 5      | 23      |
| o71      | 10      | 8      | 5      | 23      |
| o76      | 10      | 8      | 5      | 23      |
| o77      | 10      | 8      | 5      | 23      |
| o78      | 10      | 8      | 0      | 18      |
| o79      | 10      | 8      | 0      | 18      |
| o86      | 10      | 8      | 5      | 23      |
| o87      | 10      | 8      | 5      | 23      |
| o92      | 10      | 8      | 5      | 23      |
| o93      | 10      | 8      | 5      | 23      |
| o94      | 10      | 8      | 0      | 18      |
| o95      | 10      | 8      | 0      | 18      |
| o166     | 0       | 0       | 5      | 5       |
| o167     | 0       | 0       | 5      | 5       |
| o172     | 0       | 0       | 5      | 5       |
| o173     | 0       | 0       | 5      | 5       |
| o174     | 0       | 0       | 0       | 0       |
| o175     | 0       | 0       | 0       | 0       |
| o182     | 0       | 0       | 5      | 5       |
| o183     | 0       | 0       | 5      | 5       |
| o188     | 0       | 0       | 5      | 5       |
| o189     | 0       | 0       | 5      | 5       |
```

```
Command Prompt - python r x + v
+-----+-----+-----+-----+-----+
| o172     | 0       | 0       | 5      | 5       |
| o173     | 0       | 0       | 5      | 5       |
| o174     | 0       | 0       | 0       | 0       |
| o175     | 0       | 0       | 0       | 0       |
| o182     | 0       | 0       | 5      | 5       |
| o183     | 0       | 0       | 5      | 5       |
| o188     | 0       | 0       | 5      | 5       |
| o189     | 0       | 0       | 5      | 5       |
| o190     | 0       | 0       | 0       | 0       |
| o191     | 0       | 0       | 0       | 0       |
| o230     | 0       | 0       | 5      | 5       |
| o231     | 0       | 0       | 5      | 5       |
| o236     | 0       | 0       | 5      | 5       |
| o237     | 0       | 0       | 5      | 5       |
| o238     | 0       | 0       | 0       | 0       |
| o239     | 0       | 0       | 0       | 0       |
| o246     | 0       | 0       | 5      | 5       |
| o247     | 0       | 0       | 5      | 5       |
| o252     | 0       | 0       | 5      | 5       |
| o253     | 0       | 0       | 5      | 5       |
| o254     | 0       | 0       | 0       | 0       |
| o255     | 0       | 0       | 0       | 0       |
+-----+-----+-----+-----+-----+

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 4
Two randomly selected feasible objects are o87 and o230,
and o230 is strictly preferred over o87.

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 5
All optimal objects: o174, o175, o190, o191, o238, o239, o254, o255
```

```
Command Prompt - python r x + v
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 5
All optimal objects: o174, o175, o190, o191, o238, o239, o254, o255

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 6

Choose the preference logic to use:
1. Penalty Logic
2. Qualitative Choice Logic
3. Exit

Your Choice: 1

You picked Penalty Logic
Enter Preferences File Name: testpenaltylogic.txt

Choose the reasoning task to perform:
1. Encoding
2. Feasibility Checking
3. Show the Table
4. Exemplification
5. Omni-optimization
6. Back to previous menu

Your Choice: 6

Choose the preference logic to use:
1. Penalty Logic
2. Qualitative Choice Logic
3. Exit

Your Choice: 44

Wrong choice! Please your choice:
```