Major Course Output 2 (Updated 10/22/23) **Release Date:** October 6, 2023
ChatBot **Due Date:** November 26, 2023 (Sun), 11:59 PM

1 Project Description

In this project, students will apply the concepts and theories they have learned in the course to develop a chatbot that can understand family relationships. Through this project, students should be able to demonstrate the following learning outcomes:

- **LO3.** Collaboratively build and evaluate an expert system prototype that addresses a need of a particular user group, community, or organization.
- **LO5.** Articulate ideas and present results in correct technical written and oral English.

2 Project Specifications

This section contains the specifications for this major course output.

2.1 Overview

In this project, students will make use of a logic-based inference system as the backbone of a simple conversational chatbot that can understand family relationships and answer queries about them.

2.2 Inference Engine

Note: This section has been updated, due to issues that were reported with the previous `logic.py` inference engine. The said file is still included in the package for you to play around with, but you are NOT required to use it anymore.

For this project, students will use PROLOG as the inference engine for the chatbot. PROLOG is a logic programming language that can do the following:

- Represent a set of facts and rules in a knowledge base.
- Make a query as to whether a certain formula can be derived from the knowledge base or not.

PROLOG can be downloaded here.

This is an example of a PROLOG knowledge base:

```
dog(rottweiler). % rottweiler is a dog
cat(munchkin). % munchkin is a cat
animal(A) :- cat(A). % for all A, A is a cat implies A is an animal
animal(A) :- dog(A). % for all A, A is a dog implies A is an animal
```

From this knowledge base, you can make the following query:

```
animal(munchkin)
```

This will return true, indicating that munchkin is an animal.

Furthermore, we can make this query:

```
animal(X)
```

This will return `X=munchkin` and `X=rottweiler`, because these are the possible unifications that will make the formula entailed by the knowledge base. In other words, munchkin and rottweiler are the animals in the knowledge base.

There are some important things to note about PROLOG. PROLOG follows a **closed-world assumption**, which means that for every query, it only determines whether it is entailed by the knowledge base (true), or not entailed by the knowledge base (false). A false result could mean either contingency or contradiction, but there's no way to tell which one. Therefore, you may need to define additional predicates to capture some of the required logic for the chatbot, especially for the parts where you have to check whether a statement is feasible or not.

There is much more to PROLOG than just this. There are many tutorials about PROLOG on the web, such as this crash course. Feel free to explore them.

For this project, students will write their chatbot in Python. The Python program should **interface with the PROLOG engine** for the inference. This can be done through a Python package called `pyswip`, which allows you to call PROLOG from Python. More details about `pyswip`, including example codes, can be found here. Note that your the interface of your chatbot, including the processing of the user's prompts, should be done on the Python side. Thus, there is no need to worry about how to get input or display output from PROLOG. The purpose of PROLOG is merely to serve as an inference engine for the intelligence of the bot. In other words, the flow of your Python program should be:

1. Show some welcome message if you want.
2. Get user's prompt.
3. Process the sentence to determine the user's intention.
4. Call PROLOG functions to process the input.
   (a) If the input is a statement, query PROLOG to check the fact is feasible. If yes, add it to the knowledge base. If not, display some error message.
   (b) If the input is a question, query PROLOG to get the answer. Display the answer to the user.
5. Repeat.

More details about the user prompts can be found in the next section.

2.3 The Chatbot

For this project, you are to develop a conversational chatbot that can understand a set of statements and sentence patterns. The user should be able to communicate with the chatbot by giving it a prompt. After which, the chatbot responds accordingly. There should be two kinds of prompts: a statement or a question. Statements allow the user to tell the chatbot new pieces of information. On the other hand, questions allow the user to ask the chatbot certain questions. **To get full credit, make sure that you follow the sentence patterns below exactly.**

The boxes in the sentence patterns below can be replaced with the name of a person. For simplicity, in this project, we will only consider names that do not have spaces and only contain letters. You may also assume that names will always be written such that the first letter is capitalized, while the rest is lowercase. Furthermore, you may assume that names are unique. No two people will have the same name.

2.3.1 Statement prompts

Statement prompts can be one of the following sentence patterns:

Sentence Pattern	
☐ and ☐ are siblings.[1]	☐ is a brother of ☐.
☐ is a sister of ☐.	☐ is the father of ☐.
☐ is the mother of ☐.	☐ and ☐ are the parents of ☐.
☐ is a grandmother of ☐.	☐ is a grandfather of ☐.
☐ is a child of ☐.	☐, ☐ and ☐ are children of ☐.[2]
☐ is a daughter of ☐.	☐ is a son of ☐.
☐ is an uncle of ☐.	☐ is an aunt of ☐.

The chatbot should respond to a statement prompt as follows. If the information is feasible given the current known facts, the chatbot should acknowledge that it learned something and add the new piece of information in the knowledge base. If the information contradicts with the known facts, the chatbot should tell the user that that is impossible, and it won't add it to the knowledge base.

[1]Sibling means at least one parent must be the same.
[2]Can be two or more children

2.3.2 Question Prompts

Question prompts always end with a question mark (?) and can be one of the following sentence patterns:

Sentence Pattern	
Are ☐ and ☐ siblings?	Who are the siblings of ☐?
Is ☐ a sister of ☐?	Who are the sisters of ☐?
Is ☐ a brother of ☐?	Who are the brothers of ☐?
Is ☐ the mother of ☐?	Who is the mother of ☐?
Is ☐ the father of ☐?	Who is the father of ☐?
Are ☐ and ☐ the parents of ☐?	Who are the parents of ☐?
Is ☐ a grandmother of ☐?	Is ☐ a grandfather of ☐?
Is ☐ a daughter of ☐?	Who are the daughters of ☐?
Is ☐ a son of ☐?	Who are the sons of ☐?
Is ☐ a child of ☐?	Who are the children of ☐?
Are ☐, ☐ and ☐ children of ☐?	Is ☐ an aunt of ☐?
Is ☐ an uncle of ☐?	Are ☐ and ☐ relatives?

The chatbot should respond to a question prompt by providing the appropriate answer. For yes/no questions, the chatbot should answer either "yes", if the knowledge base can prove it, or "no", if it cannot.

2.3.3 Chatbot Implementation

You should implement the chatbot as a Python program. Graphical user interface (GUI) is allowed, but not required. Do not put any of the inference logic in the Python program. In general, the main tasks in this project are as follows:

1. Encoding the relationship rules into the knowledge base in a PROLOG file.
2. Parsing the input prompts and translating them into the appropriate PROLOG queries.
3. Making the appropriate queries to the knowledge base in order to update the facts or answer the questions.

If invalid prompts are provided (i.e., does not follow any of the given sentence patterns), the program should not crash and should respond instead with an error message. You are free to add additional prompts beyond the ones given, as long as they do not conflict with any prompts in the list.

You must make sure that the chatbot behaves intelligently. That is, it should be able to infer facts based on real-world concepts even if they were not explicitly stated. The following are some examples of intelligent behavior that should be handled by the chatbot:

Example 1:

```
> Bob is the father of Alice.
OK! I learned something.
> Alice is the mother of John.
OK! I learned something.
> Is Bob a grandfather of John?
Yes!
```

In this example, even though it wasn't explicitly stated, the chatbot inferred the grandfather relationship.

Example 2:

```
> Mark is the father of Patricia.
OK! I learned something.
> Mark is a daughter of Ann.
That's impossible!
```

In this example, Mark couldn't be a daughter of Ann, because Mark is male (since he was already established to be a father).

Example 3:

```
> One is the father of Two.
OK! I learned something.
> Two is the father of Three.
OK! I learned something.
> Three is the father of One.
That's impossible!
```

In this example, the given set of relationships is just not possible in real life.

Example 4:

```
> Lea is the mother of Lea.
That's impossible!
```

Lea couldn't have given birth to herself.

Note that there are other rules that you should capture apart from these examples. Part of this project is for you to spend some time thinking deeply on how real-world knowledge is translated into first order logic. To get full credit, your chatbot should be able to exhibit many cases of intelligent inference.

3 Report

In addition to the chatbot program, you are required to write a report documenting the implementation of the project. At the minimum the report should contain the following:

1. **Brief Introduction:** A short introduction describing the chatbot that was developed. Please make this section short and concise.
2. **Knowledge Base:** A section listing the formulas that were encoded into the knowledge base. You must enumerate and describe each formula. Use the standard first order logic notation when writing the formulas.
3. **Chatbot Implementation**: A section describing how the chatbot was implemented, including how it parses the prompts and how it queries the knowledge base to provide intelligent answers.
4. **Results**: A section describing the chatbot in action, highlighting its capabilities and aspects of intelligence as well as identifying its weaknesses.
5. **Limitations and Challenges**: A discussion on the limitations of the chatbot that was developed, especially when compared to large language model (LLM) based chatbots like ChatGPT. What are the challenges that make these limitations difficult to overcome?
6. **Conclusion:** A conclusion section summarizing the process of implementing the project, including reflections, realizations, and insights obtained.
7. **Table of Contributions:** A table showing the contributions of each member to the project.

There is no required format for the report, but please refrain from submitting very long reports with little substance. Please minimize copying or paraphrasing already known concepts and theories to make the report longer. The bulk of the report should contain the group's personal ideas and insights.

4 Deliverables

There are two deliverables for this project: a zip file containing the source files for the project, and a pdf file containing the report. The source files should include:

1. All the source code and files needed to run the project.
2. An `instructions.txt` file which contains instructions on how to run the program, as well as any other non-trivial information needed to operate the program properly.

The file name of the zip file should be: `mco2_<surname 1>_<surname 2>_<surname 3>_<surname 4>.zip` with the surnames in alphabetical order. The file name of the report should be `mco2_<surname 1>_<surname 2>_<surname 3>_<surname 4>.pdf` with the surnames in alphabetical order.

5 Academic Honesty

Honesty policy applies. Please take note that you are **NOT** allowed to borrow and/or copy-and-paste in full or in part any existing related program code from the internet or other sources (such as printed materials like books, or source codes by other people that are not online). **Violating this policy is a serious offense in De La Salle University and will result in a grade of 0.0 for the whole course**.

Please remember that the point of this project is for all members to learn something and increase their appreciation of the concepts covered in class. Each member is expected to be able to explain the different aspects of their submitted work, whether part of their contributions or not. **Failure to do this will be interpreted as a failure of the learning goals, and will result in a grade of 0 for that member**.