UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

First-order Logic (FOL)

Introduction to Artificial Intelligence

TRIỆU NHẬT MINH - 21127112
BÙI ĐỖ DUY QUÂN - 21127141
HOÀNG ĐỨC VIỆT - 21127203

Lecturers:

Nguyễn Tiến Huy

Nguyễn Trần Duy Minh

16th May 2023

Contents

1 Self-assessment 2
2 Working with the Prolog tool 2
2.1 Main features of Prolog language 2
2.2 How to implement Prolog language 3
2.3 Build a Knowledge Base with Prolog 7
3 Implement logic deductive system in the programming language . . 8
4 References 9

1 Self-assessment

Criteria	Completion
Working with Prolog tools	100%
Build a Knowledge Base with Prolog	100%
Implement logic deductive system in the programming language (Python)	100%

2 Working with the Prolog tool

2.1 Main features of Prolog language

Prolog (or PROgramming in LOGics) is a programming language that uses logic to express programs. Prolog is a declarative language, which means that a program consists of KBs on the facts and rules (Logical relationship) rather than computing how to find a solution. The program consists of facts and rules that define relations between entities. A computation is performed by asking a query about these relations.

Prolog has several distinctive features, such as its declarative meaning, its built-in mechanism for logical inference, and its natural handling of lists and recursion. It can also try different options (backtracking) until it finds the solution.

Examples:

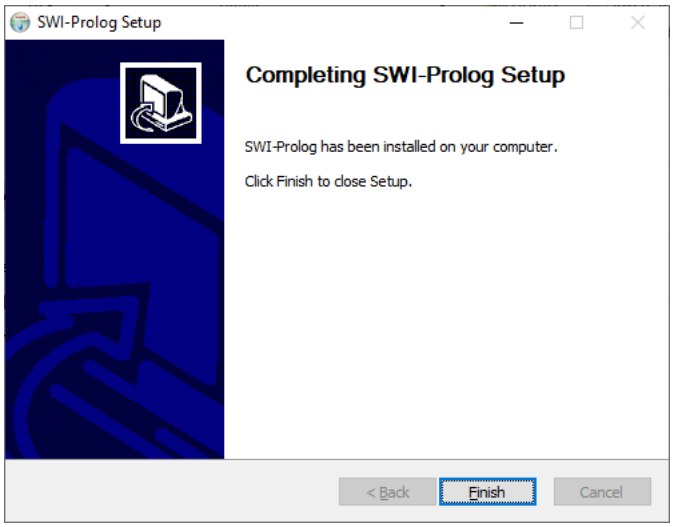
English sentence	FOL knowledge base	Prolog implementation
Everyone loves Santa.	∀x (LOVE(x,Santa))	love(X, "Santa").
Rudolph is a reindeer and Rudolph has a red nose.	REINDEER(Rudolph) ∧ REDNOSE(Rudolph)	reindeer("Rudolph"), rednose("Rudolph").
John owns a dog.	∃x (DOG(x) ∧ OWN(John,x))	dog(_) , own("John", _).
Anyone who does not play is not a football star.	∀x (¬ PLAY(x) → ¬ STAR(x))	not(star(X)) :- not(play(X)).
Everyone who know Prolog or know Python is smart.	∀x (KNOW(x, Prolog) ∨ KNOW(x, Python) → SMART(x))	smart(X) :- (know(X, "Prolog") ; know(X, "Python")).

where the given predicates are:

- LOVE(x, y): x loves y.
- REINDEER(x): x is a reindeer.
- REDNOSE(x): x has a red nose.
- DOG(x): x is a dog.
- OWNS(x, y): x owns y.
- PLAY(x): x plays.
- STAR(x): x is a football star.
- KNOW(x, y): x knows y.
- SMART(x): x is smart.

2.2 How to implement Prolog language

- Our possible way to implement Prolog language on a programming environment is to use SWI-Prolog. To use SWI-Prolog, one needs to install it and then write Prolog programs using any text editors or IDEs. The programs can be compiled and executed (consulted) by using the swipl command or the graphical user interface (GUI) of SWI-Prolog.
- Steps to install SWI-Prolog (Windows):
  1. Go to SWI-Prolog download page and click **Stable release**.
  2. Choose the suitable version.
  3. Choose **I understand** and download the installation.
  4. Install until this popup appears on the screen, which means you installed successfully.



- Prolog syntax:

Operator	Prolog syntax
and	,
or	;
!=	\=
if	:-
not	not

  - Variable name starts with a capital letter. Predicate and function name starts with a letter.

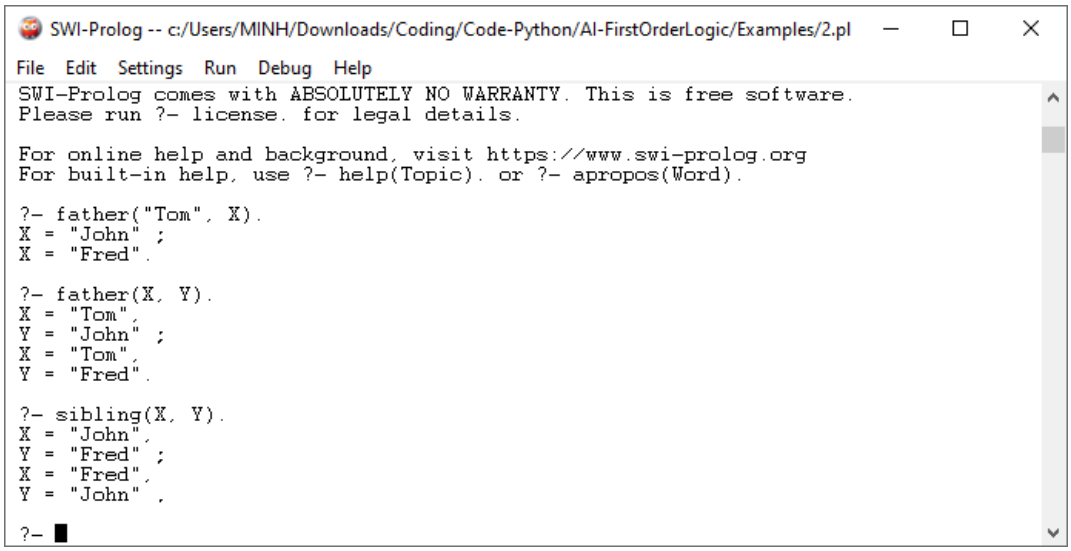
- All sentences must end with a period (.).
- When defining a fact, if you do not declare a specific variable, by default, the program understands that variable has the value "for all".
  For example: **loves(X, john)**. Which means, everyone loves john.
- Syntax for Prolog rules: **left_side :- right_side.** /* left_side ← right_side.*/. Notice that **left_side** expression is required to contain only 1 positive literal.
- Syntax for Query: **?- query.** . If the program finds the answer to the query or proves the user's query is correct, SWI-Prolog will display **true** or the answers. Otherwise, SWI-Prolog returns **false**.

- Examples
  The following examples below are located in Examples/x.pl where x is its order.

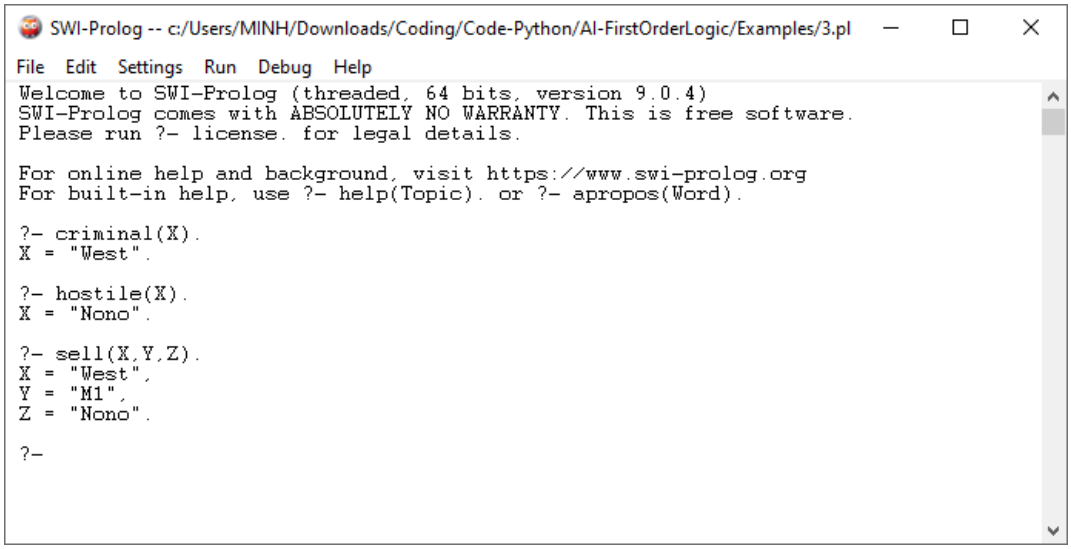
1. Given three animals: Buffalo (named Bob), pig (named Pat), and slug (named Steve), and a knowledge base: if A is faster than B (faster(A, B)) and B is faster than C (faster(B, C)) then A is faster than C (faster(A, C)). The fact is Bob is faster than Pat, Pat is faster than Steve. Find the animals that are faster than Steve, Pat, and Bob.

   The last query *faster(X, "Bob")* returns error message because Bob is the faster of the three, so there is no animal faster than Bob and the program reach the stack memory limitation due to infinity loop.

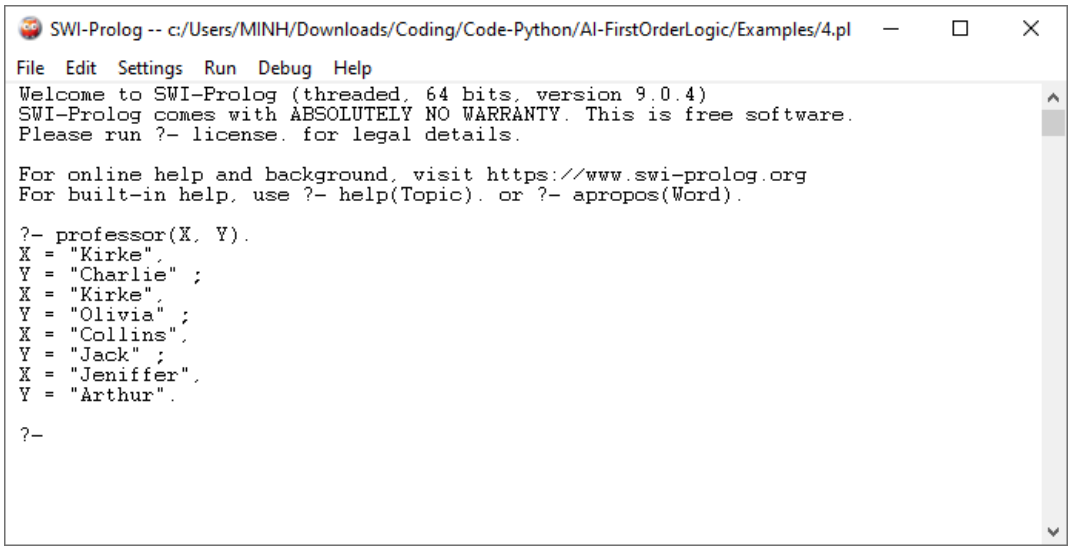
2. Given the knowledge bases: P is father of C if P is parent of C and P is male. P1 and P2 are siblings if they have same parent. The program checks who is Tom's child, list paternities and the sibling relationships.



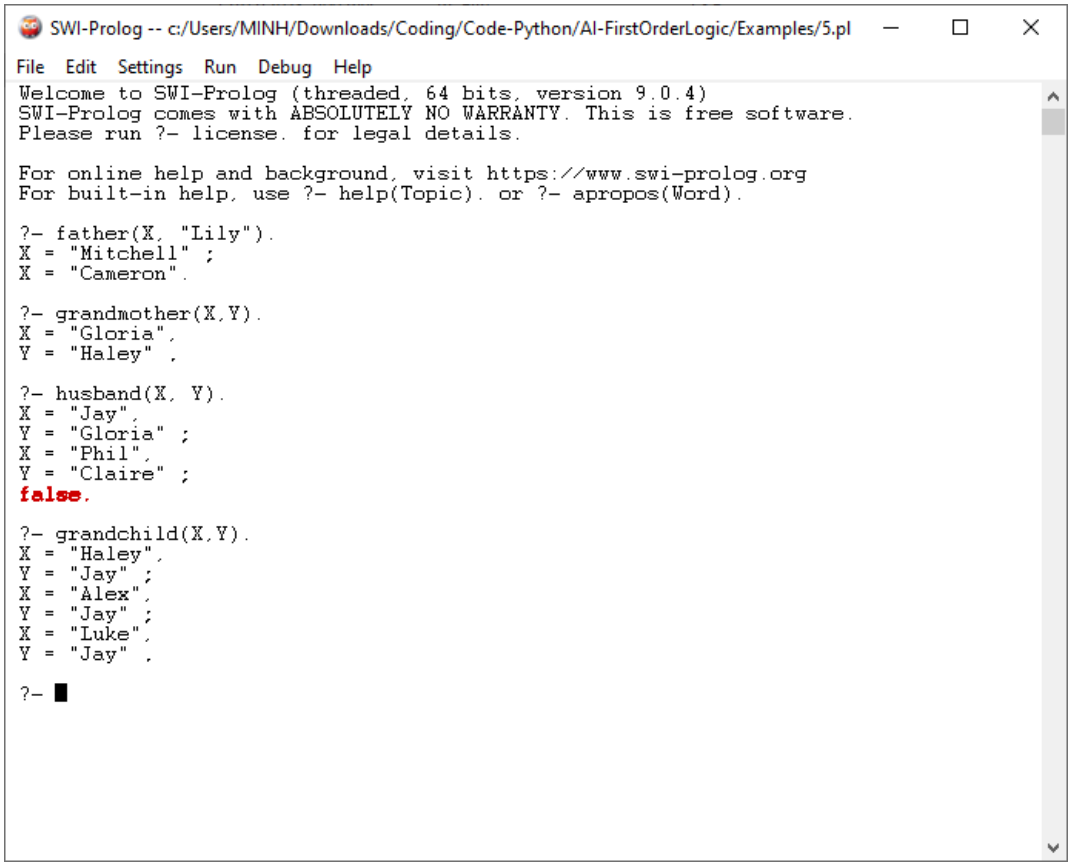
3. The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West. Who is the criminal.



4. Given four students, three teachers and three courses. Find the professor and the student of each course.



5. Given a family, inspired by the "Modern Family" sitcom. Find out the relationship between some members.



The *husband(X, Y).* query returns a false result because there are no more couples in the data but the user tries to explore more.

2.3 Build a Knowledge Base with Prolog

- **Set of 20 questions to ask the given family tree of the British Royal family:**

1. Who is Prince Andrew's mother?
2. Was Queen Elizabeth the wife of Mia Grace Tindall?
3. List all married couples.
4. List all divorced couples.
5. List grandchildren of Prince Philip.
6. Who is the grandparent of Princess Charlotte?
7. Who is the sister of Prince Harry?
8. Who is the sister of Princess Beatrice?
9. Who is the brother of Princess Edward?
10. Who is the uncle of Prince George?
11. Who is Prince Charles's sibling?
12. Who is the uncle of Princess Beatrice?
13. Who is son of Prince Phillip?
14. Who is the nephew of Prince Andrew?
15. Who is the niece of Princess Anne?
16. Who is the wife of Prince Edward?
17. Who is the grandson of Prince Phillip?
18. Who is the child of Prince Phillip?
19. Who is the daughter of Prince Phillip?
20. Who is the grandmother of grandmother Mia Grace Tindall?

The questions and Prolog queries are written in *BritishRoyalFamily-Q.txt* file.

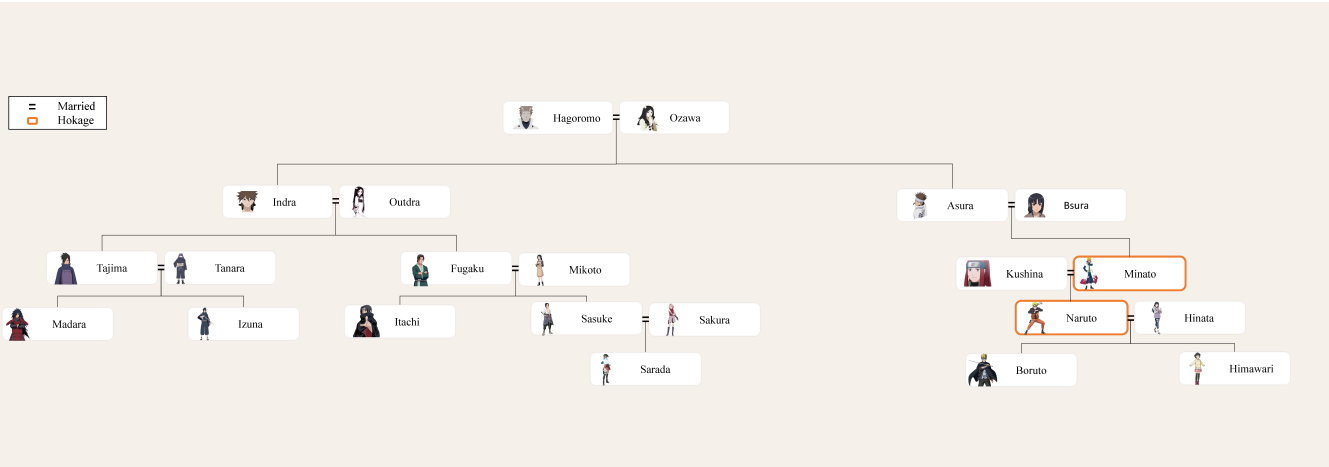
- **Set of 20 questions to ask the newly constructed knowledge system:**

1. Who is parent of Boruto?
2. Who is father of Sarada?
3. Is Hinata male?
4. Who is grandparent of Himawari?
5. Who is grandchild of Indra?
6. Who is sibling of Sasuke?
7. List all married couples.
8. List all grandchild of Hagoromo
9. Who is uncle of Sarada?
10. Who is son of Naruto?
11. Who is daughter of Sakura?
12. Who is child of Tajima?
13. Is Sarada a daughter of Naruto?
14. Who is nephew of Tajima?
15. Who is grandmother of Kushina?
16. Who is granddaughter of Bsura?
17. Who is the grandchild of Outdra?
18. Is Naruto sibling of Itachi?

19. Who is the aunt of Mikoto?
20. List all hokage of Konoha.

The questions and Prolog queries are written in *MyBase-Q.txt* file.

The figure below shows the diagram of the relationship between the objects in the selected topic.



3 Implement logic deductive system in the programming language

- Our group decided to implement the logic system by using **Python** programming language.
- The reasoning method that we use is forward chaining.
- We do have the Prolog file, which contains the data (facts and rules).
- At first, the program requires the input that is the name of file containing KBs and questions.
- Then, the **converter** function will read that file and transfer all the data into a Python file named **PrologPython.py**:
  - First the system will read each line of the facts and relationships into a string by calling **Converter** function.
  - The data will be converted to suitable data types (lists containing names of males, females, parent's relationship, or married's relationship) or suitable functions (such as, **husbandf(X.Y)** to find relationships related to "Husband", or **fatherf(X,Y)** to find relationships related to "father", etc.
  - After defining the facts and relationships, the **Converter** function writes the string into the **PrologPython.py** file.

- After gaining all the data, the program will read the question and return the output is the answer (true/false/specific name), an input question and an output answer are similar to **SWI-Prolog**:
  - Our program will try each of the data into the case.
  - For example, to find who is Naruto's father, we will list all the male characters that have children. Then the program will try to find the male character that has a child whose name is Naruto.
- Finishing answers one question, the users can continue typing another question, or they can choose to exit the program by typing **exit.** command.

4 References

- Prolog main features
- Prolog fundamentals
- First-order logic examples
- First-order logic lecture slide