CZ3005 Artificial Intelligence - Assignment 2

Learning to Use Prolog as a Logic Programming Tool

Learning Objective:	To reinforce the understanding on the use of Prolog as a tool of logic programming and the specification of declarative knowledge – semantic information.
Learning Outcome:	a) Understand the separation of Knowledge from the inference mechanism in Knowledge-Based System (KBS). b) Understand that the proof procedure is based on the refutational proof using resolution on Horn Clauses. c) Appreciate the ordering of the clauses of the same class affects the proof sequence.
Tools:	SWI-Prolog - http://www.swi-prolog.org/

Learning Activities:	1. Download and install SWI-Prolog.
	 2. Watch the two briefing videos on NTULearn:
	     a. "**Prolog (1.Introduction).mp4**"
	     b. "**Prolog (2.Building a KB).mp4**".
	     These two videos explain the details in the guide on "**How to Program in Prolog.pdf**".

Deliverables:	This assignment consists of two problems. You are to specify each of them as a declarative knowledge base in Prolog.
Team Formation	You are required to form a team of up to 3 people to complete this project (feel free to continue working with the same team as in Lab 1). A team with only a single person is also allowed. For team size that is larger than one, you need to clearly state the contribution of each team member.

Submission Instruction:	Complete both Exercise 1 and Exercise 2.
	
	 Your submission should contain:
	   1) The Prolog knowledge bases (A "<team_name>_qn_1_2.pl" file for Exercise 1.2, a "<team_name>_qn_2_1.pl" file for Exercise 2.1 and a "<team_name>_qn_2_2.pl" file for Exercise 2.2), and
	   2) A brief report in PDF format "<team_name>_Assignment3.pdf" showing the:
	       a. FOL statements (where required by the question),
	       b. Traces of the queries for the two questions.
	
	 You should include your name, tutorial group number (e.g., TSR1) and matriculation card number in the cover page of your report.
	
	 Submit your report 14 days after your lab session by emailing it to your TA (Information can be found on NTULearn under the description of Assignment 2).

[Quick References]

Key syntax of Prolog:

1. Defining a fact: **P(X).**

   For example: male(peter). "Peter is a male".

2. Defining a relationship: **R(X,Y).**

   For example, sister(anne, mary). "Anne is a sister of Mary".

3. Defining a conjunctive (AND) sentence: **Goal :- Premise 1, Premise 2, …, Premise N.**

   For example, mother_of(X,Y) :- child_of(Y,X), female(X), X \= Y. "X is the mother of Y if Y is the child of X and X is female and X is not Y".

4. Defining a disjunctive (OR) sentence: **Goal :- Premise 1; Premise 2; …; Premise N.**

   For example, lose_weight(X) :- eat_less(X); exercise_more(X). "X loses weight if X eats less or X exercises more".

5. How to produce a **trace**:

   In the Prolog command line interface, type **trace,** before entering your query. Then, press **<enter>** to trace step-by-step. For example:

```
?- edit('family.pl').
true.

?- trace,parent_of(warren,X).
   Call: (11) parent_of(warren, _10898) ? creep
   Exit: (11) parent_of(warren, jerry) ? creep
X = jerry ;
   Redo: (11) parent_of(warren, _10898) ? creep
   Exit: (11) parent_of(warren, kather) ? creep
X = kather ;
   Redo: (11) parent_of(warren, _10898) ? creep
   Call: (12) brother(_10898, _13274) ? creep
   Exit: (12) brother(jerry, kather) ? creep
   Call: (12) parent_of(warren, kather) ? creep
   Exit: (12) parent_of(warren, kather) ? creep
   Exit: (11) parent_of(warren, jerry) ? creep
X = jerry ;
   Redo: (12) parent_of(warren, kather) ? creep
   Call: (13) brother(kather, _14372) ? creep
   Fail: (13) brother(kather, _14416) ? creep
   Fail: (12) parent_of(warren, kather) ? creep
   Redo: (12) brother(_10898, _14504) ? creep
   Exit: (12) brother(peter, warren) ? creep
   Call: (12) parent_of(warren, warren) ? creep
   Call: (13) brother(warren, _14636) ? creep
   Fail: (13) brother(warren, _14680) ? creep
   Fail: (12) parent_of(warren, warren) ? creep
   Fail: (11) parent_of(warren, _10898) ? creep
false.
```

Exercise 1: The Smart Phone Rivalry (15 marks)

sumsum, a competitor of appy, developed some nice smart phone technology called galactica-s3, all of which was stolen by stevey, who is a boss of appy. It is unethical for a boss to steal business from rival companies. A competitor of is a rival. Smart phone technology is business.

1. Translate the natural language statements above describing the dealing within the Smart Phone industry in to First Order Logic (FOL).

(5 marks)

2. Write these FOL statements as Prolog clauses.

(5 marks)

3. Using Prolog, prove that Stevey is unethical. Show a trace of your proof.

(5 marks)

Exercise 2: The Royal Family (10 marks)

The old Royal succession rule states that the throne is passed down along the male line according to the order of birth before the consideration along the female line – similarly according to the order of birth. queen elizabeth, the monarch of United Kingdom, has four offsprings; namely:- prince charles, princess ann, prince andrew and prince edward – listed in the order of birth.

1. Define their relations and rules in a Prolog rule base. Hence, define the old Royal succession rule. Using this old succession rule determine the line of succession based on the information given. Do a trace to show your results.

(5 marks)

2. Recently, the Royal succession rule has been modified. The throne is now passed down according to the order of birth irrespective of gender. Modify your rules and Prolog knowledge base to handle the new succession rule. Explain the necessary changes to the knowledge needed to represent the new information. Use this new succession rule to determine the new line of succession based on the same knowledge given. Show your results using a trace.

(5 marks)

[**Important**] *Prolog interprets capital lettered words as variables. Thus, in your Prolog code, please write the above terms in blue using all lower-case letters.*

- End of Assignment 2 -