Introduction to logic programming

Instructions

1. Download the .zip file from the extranet, unzip, and import in Eclipse.
2. Do the exercises in the proposed order!
3. For each exercise: run it, watch the result and try and understand what is happening.
4. Do not hesitate to modify the code when it helps you understand what it does. The original is still there in the .zip file on the extranet.
5. Answer the questions.

Places to find help:

- The extranet contains useful documents.
- Google.
- Your colleagues but beware of the noise.
- Try and guess answers that make sense. When stuck, move along and do not waste time.

Ex 1

1. Open up famille.prolog in Eclipse and take a look at its content.
2. Start Prolog.scala.
3. Load the content of family.prolog :

   ```
   prolog> :l famille.prolog
   ```

4. What is the result of: `prolog> ?child(robert,fred).`

5. `prolog> ?child(robert,robert).`

6. `prolog> ?child(robert,X).`

7. In fact the previous command returns several results. Use this command several times to get them all:

   `prolog> ?more.`

8. `prolog> ?female(X).` (Beware there are several results, use `?more.` to catch them all).

9. Let's add a new concept:

   `prolog> male(X) :- not(female(X)).`

   According to this definition, what does this return: `prolog> ?male(robert).`

10. The not operator is special in Prolog: not(P) succeeds when P fails. However this can be problematic sometimes. For example, try this: `prolog> ?male(X).`

11. Let's add the concept of siblings:

    `prolog> sibling(X,Y) :- child(X, Z), child(Y, Z).`

    According to this definition, what does this return: `prolog> ?sibling(pierre,robert).`

12. `prolog> ?sibling(robert,jeanne).`

13. `prolog> ?sibling(robert,X).` (Use `?more.` to get all the results).

14. It seems that Robert is his own brother! Why is that happening?

15. To correct this, the sibling rules need to change. Warning: when a rule is added into the system, it stays there forever and cannot be removed. So you need to re-start Prolog.scala and reload family.prolog before entering these new rules:

```
prolog> same(X,X).
prolog> sibling(X,Y) :- child(X,Z), child(Y,Z), not(same(X,Y)).
```

What is the result of: `prolog> ?sibling(robert,X).` (Use `?more.` to get all the results) ?

16. Robert is no more his own brother, however Pierre and Sylvie appears twice. Why is that happening?

17. How can you solve this problem? Hint: only consider the mothers or the fathers. Figure out a new rule and don't forget to reload otherwise the older rules would still apply.

18. Let's add the concepts of parents and ancestors :

```
prolog> parent(X,Y) :- child(Y,X).
prolog> ancestor(X,Y) :- parent(X,Y).
prolog> ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).
```

Try out several requests using these rules:

19. Add the concept of X is the uncle of Y.

20. Add the concept of X is the cousin of Y.