Logic Programming Language (III)

Rule (cont.)
- Exercise:
  - Given the facts that:
    
    ```
    likes(john, mary).
    likes(john, trains).
    likes(peter, fast_cars).

    hobby(john, trainspotting).
    hobby(tim, sailing).
    hobby(helen, trainspotting).
    hobby(simon, sailing).
    ```

  - Ask students to write a rule that if person1 and person2 have the same hobby, person1 likes person2.
  
  - Solution:
    ```
    likes(Person1, Person2) :- hobby(Person1, Hobby), hobby(Person2, Hobby).
    ```

  - Show exercise1.pl

  - Query:
    ```
    likes(john, trains). // true
    likes(helen, john). // true
    likes(tim, helen). // true
    likes(john, helen). // true
    ```

Recursion
- Recursion is a way to loop in Prolog.
- It allows repeatedly execute some operation till a certain point is reached or over a whole data structure.
- A recursion should have a first fact that acts as the base case.
- Then it should have some rule(s) that performs some recursive operation.

- Example1 (grandcanyon.pl)
  - To determine if there is a route to Grand Canyon.
    ```
    on_route(grand_canyon).
    on_route(Place) :- move(Place, _Method, NewPlace), on_route(NewPlace).

    move(home, bus, boston).
    move(boston, plane, las_vegas).
    move(las_vegas, vechicle, grand_canyon)
    ```

  - Query
    ```
    on_route(home).
    ```
Example 2 (ancestor.pl)

- Given

  parent(john, paul). /* paul is john's parent */
  parent(paul, tom). /* tom is paul's parent */
  parent(tom, mary). /* mary is tom's parent */

- Ask student to write a rule to determine if X is Y's ancestor.

- Solution:

  ancestor(X, Y) :- parent(X, Y). /* someone is your ancestor if they are your parents */
  ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y). /* or somebody is your ancestor if they are the parent */

- Query:

  ancestor(john, tom).

Recursion Exercise 1 (prerequisite.pl)

- Given

  prerequisite(cs231, cs151).
  prerequisite(cs232, cs231).
  prerequisite(cs333, cs231).
  prerequisite(cs421, cs333).

- Write a rule that can find all prerequisites of a course.

- Solution:

  all_prerequisite(X, Y) :- prerequisite(X, Y).
  all_prerequisite(X, Y) :- prerequisite(X, Z), all_prerequisite(Z, Y).

- Query:

  all_prerequisite(cs421, X).