Let’s get used to Prolog!!!

CS370 Symbolic Programming
2007. 3. 13

We are going to learn ...

- Basic Prolog manipulations
- Einstein’s quiz and its solution with Prolog
- Features of Prolog
  - Debugging
  - I/O
  - Libraries
- Hint for HW1

Basic Prolog Manipulations 1/2

- SICStus Prolog is in..
  - adam: /usr/local/bin/sicstus/bin
  - SWI prolog for Windows (http://www.swi-prolog.org)

- Start / End SICStus Prolog with
  - sicstus / halt

Basic Prolog Manipulations 2/2

- How can we feed rules and facts to Prolog?
  - User mode
    - :- [user].
    - ctrl + D
  - Loading source files
    - consult(program)
    - :- [program]
    - compile(program)
    - sicstus -l program

Einstein’s Quiz

- Facts:

1: There are 5 houses in 5 different colors
2: In each house lives a person with a different nationality.
3: These 5 owners drink a certain beverage, smoke a certain brand of cigar and keep a certain pet.
4: No owners have the same pet, smoke the same brand of cigar or drink the same drink.

- Further Details:

1: The Brit lives in a red house.
2: The Swede keeps dogs as pets.
3: The Dane drinks tea.
4: The green house is on the immediate left of the white house.
5: The green house owner drinks coffee.
6: The person who smokes Pall Mall rears birds.
7: The owner of the yellow house smokes Dunhill.
8: The man living in the house right in the center drinks milk.
9: The Norwegian lives in the first house.
10: The man who smokes Blend lives next to the one who keeps cats.
11: The man who keeps horses lives next to the man who smokes Dunhill.
12: The owner who smokes Blue Master drinks beer.
13: The German smokes Prince.
14: The Norwegian lives next to the blue house.
15: The man who smokes Blend has a neighbor who drinks water.

- The question is: WHO KEEPS FISH?

Einstein’s Quiz - Solution 1/4

- Data Representation
  - Each house
    - house(Color,Nationality,Beverage,Cigar,Pet)
  - Five houses
    - [House_1st, House_2nd, ... , House_5th]

Einstein’s Quiz - Solution 2/4

- Relation
  - Member
    - member(Item, [Item | _]).
    - member(Item, [_ | Rest]) :- member(Item, Rest).
  - Order
    - order(A, B, [A, B | _]).
    - order(A, B, [_ | Tail]) :- order(A, B, Tail).
  - Near
    - near(A, B, List) :- order(A, B, List) ; order(B, A, List).

Einstein’s Quiz - Solution 3/4

- template(List) :-

```
        position(List),
        member(house(red, england, _, _, _), List),
        ...
        member(house(_, germany, _, prince, _), List),
        order(house(green, _, _, _, _), house(white, _, _, _, _), List),
        near(house(_, _, _, blend, _), house(_, _, _, _, cat),
                                                                  ).
```

The man living in the house right in the center drinks milk.
The Norwegian lives in the first house.

- position([house(_, norway, _, _, _), house(_, _, _, _, _),

```
        house(_, _, milk, _, _), house(_, _, _, _, _), house(_, _, _, _, _)]).
```

Einstein’s Quiz - Solution 3/4

- template(List) :-

  position(List),

  member(house(red, england, _, _, _), List),

  ...

  member(house(_, germany, _, prince, _), List),

  ouse(white, _, _, _, _), List),

  use(_, _, _, _, cat),

  ...

  near(house(_, _, _, blend, _), house(_, _, water, _, _), List).

- position([house(_, norway, _, _, _), house(_, _, _, _, _),

  house(_, _, milk, _, _), house(_, _, _, _, _), house(_, _, _, _, _)]).

The Brit lives in a red house.
The German smokes Prince.

Einstein’s Quiz - Solution 3/4

```
• template(List) :-
        position(List),
        member(house(red, england, _, _, _), List),
        ...
```

The green house is on the immediate left of the white house.

```
        order(house(green, _, _, _, _), house(white, _, _, _, _), List),
        near(house(_, _, _, blend, _), house(_, _, _, _, cat),
        ...
        near(house(_, _, _, blend, _), house(_, _, water, _, _), List).
```

The man who smokes Blend lives next to the one who keeps cats.
The man who smokes Blend has a neighbor who drinks water.

- p [illegible]

Einstein’s Quiz - Solution 4/4

- ownerofgoldfish(X)

  :- template(List), member(house(_, X, _, _, goldfish), List).

Prolog Features - Debugging

- trace / notrace
  - the information regarding the goal's satisfaction is displayed
  - information for all the subgoals of this goal
- spy(P) - specifies the predicate P to be traced
- nospy(P) - stops spying P

- Debugging options
  - l(leap), s(skip)
  - a(abort)
  - r <i> (retry)
  - h(help)

Prolog Features - I/O

- Predicates for I/O
  - read(X) /write(X)
  - get0(C)
  - read_line(X)
- File I/O
  - by changing current I/O stream
    - see(filename), see(user)
    - tell(filename), tell(user)
  - closing the current I/O file
    - seen, told

Prolog Features - Library

- Using SICStus libraries
  - :- use_module(library(library_name)).
  - Consult SICStus manual to see all the libraries
    - CS370 home -> resources -> manual -> the Prolog library

- Using user-defined mudules
  - :- [filename].

Hint for HW1

- \+ : ‘not’ predicate
  - \+(P) fails if the goal P has a solution, and succeeds otherwise.
  - "P is not provable".

```
\+(P) :- P, !, fail.
\+(_).
```

That’s all for today!!
Thank you :)