UCS1524 – Logic Programming

Developing a Simple shell

Session Meta Data

Author	**Dr. D. Thenmozhi**
Reviewer	
Version Number	1.2
Release Date	17 October 2022

Session Objectives

- Understanding development of simple shell in ES.

Session Outcomes

- At the end of this session, participants will be able to
  - – Develop an expert system with simple shell

Agenda

- Development of simple shell
  - Facts
  - Rules
  - Queries
- Outline of reasoning process
- Answering
  - Why questions
  - How questions

Development of an Expert System

- Steps
  - Consult actual experts for that domain and learn a great deal about it yourself.
  - Extracting some understanding of the domain from experts and literature and moulding this understanding in to a chosen knowledge-representation formalism is called the art of knowledge engineering.

Expert System in Prolog

Knowledge base to identify animals

- If it has a *fur* and says *woof*, then the animal is a **dog**.
- If it has a *fur* and says *meow*, then the animal is a **cat**.
- If it has *feathers* and says *quack*, then the animal is a **duck**.

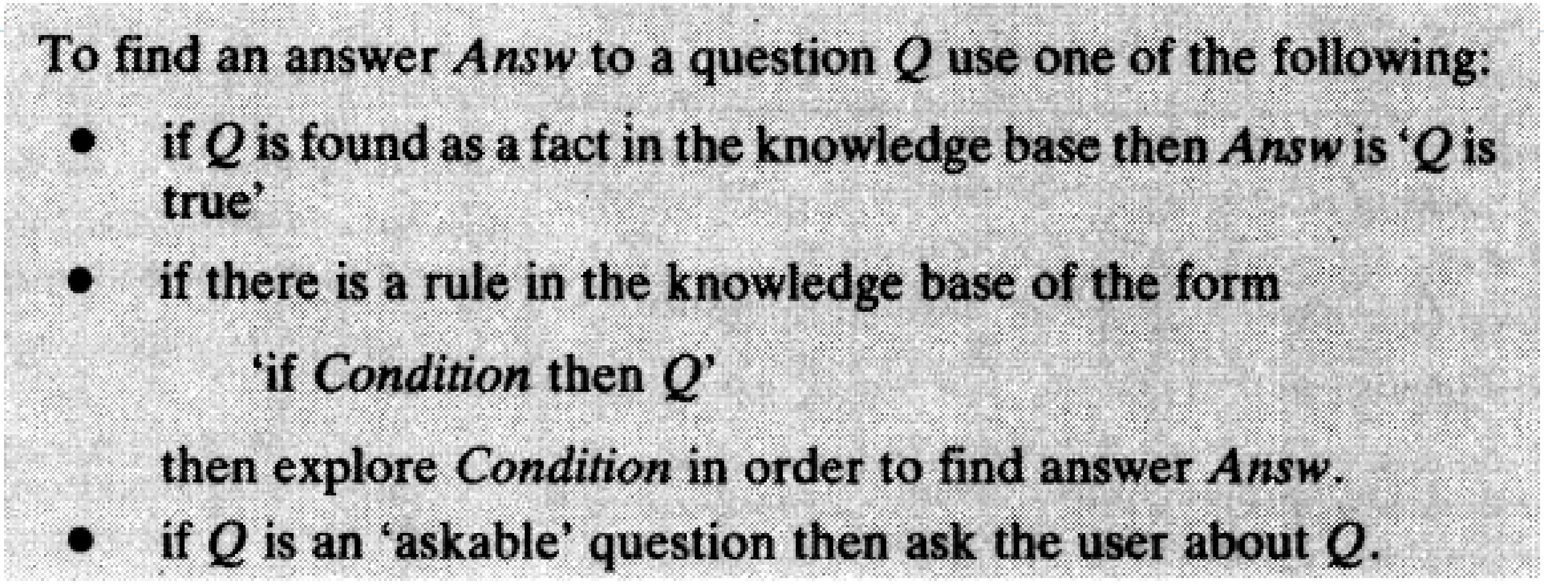
Expert System in Prolog

Knowledge base to identify animals

- If it has a *fur* and says *woof*, then the animal is a **dog**.
- If it has a *fur* and says *meow*, then the animal is a **cat**.
- If it has *feathers* and says *quack*, then the animal is a **duck**.

- animal(dog) :- is_true('has fur'), is_true('says woof').
- animal(cat) :- is_true('has fur'), is_true('says meow').
- animal(duck) :- is_true('has feathers'), is_true('says quack').
- is_true(Q) :- format("~w?\n", [Q]), read(yes).
- Goal : ?- animal(A).

Outline of a reasoning process

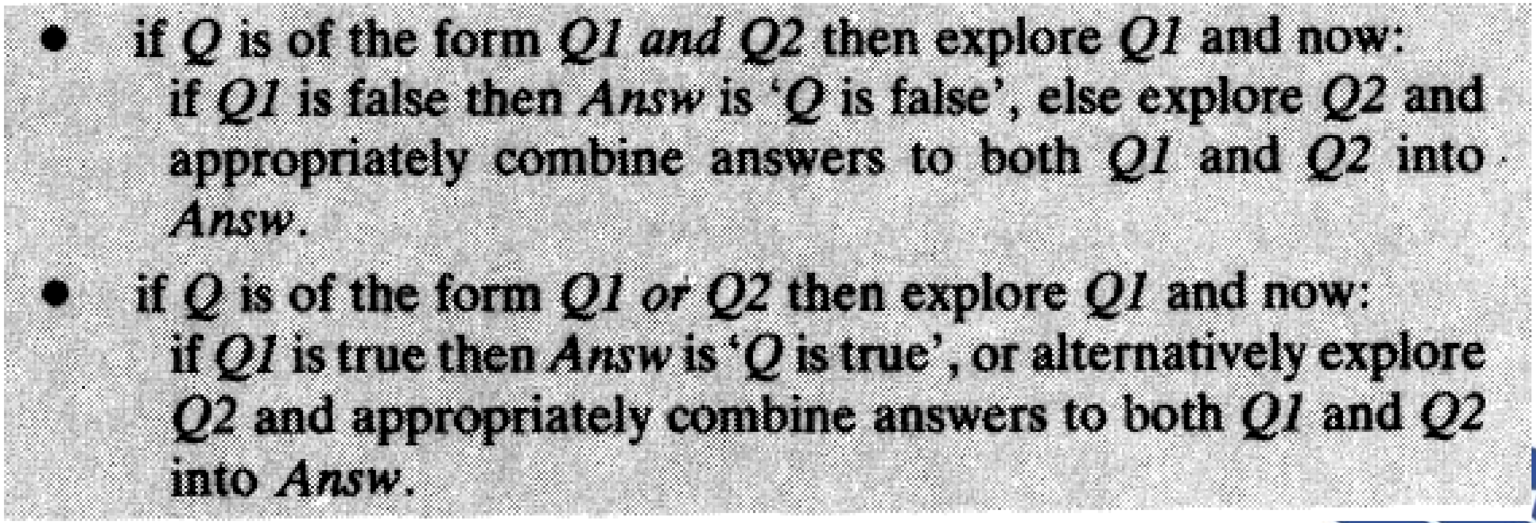


To find an answer *Answ* to a question *Q* use one of the following:

- if *Q* is found as a fact in the knowledge base then *Answ* is '*Q* is true'
- if there is a rule in the knowledge base of the form

  'if *Condition* then *Q*'

  then explore *Condition* in order to find answer *Answ*.
- if *Q* is an 'askable' question then ask the user about *Q*.



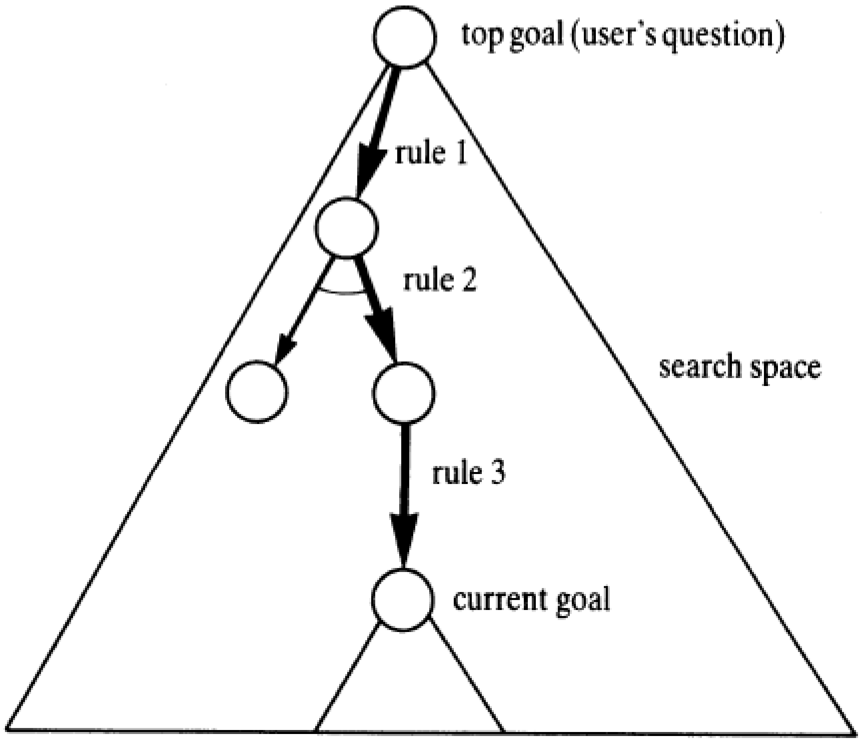
- if *Q* is of the form *Q1 and Q2* then explore *Q1* and now: if *Q1* is false then *Answ* is '*Q* is false', else explore *Q2* and appropriately combine answers to both *Q1* and *Q2* into *Answ*.
- if *Q* is of the form *Q1 or Q2* then explore *Q1* and now: if *Q1* is true then *Answ* is '*Q* is true', or alternatively explore *Q2* and appropriately combine answers to both *Q1* and *Q2* into *Answ*.

Answering why questions

- A 'why' question occurs when the system asks the user for some information and the user wants to know why this information is needed.
- Suppose that the system has asked:
  - Is a true?
- The user may reply:
  - why?
- Explanation
  - I can use a to investigate b by rule Ra, and
  - I can use b to investigate c by rule Rb, and
  - I can use c to investigate d by rule Rc, and
  - I can use y to investigate z by rule Ry, and
  - z was your original question.

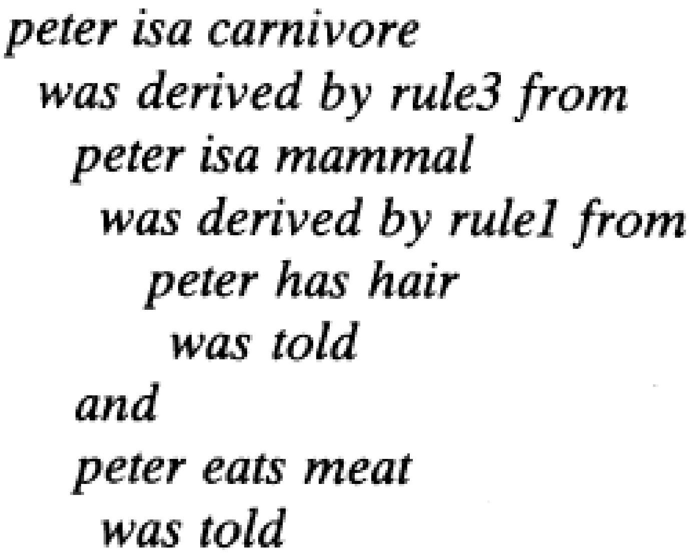
Answering why questions

- Trace



Answering how questions

- A proper way of answering how question is to display the evidence: i.e rules and subgoals from which the conclusion was reached



```
peter isa carnivore
  was derived by rule3 from
    peter isa mammal
      was derived by rule1 from
        peter has hair
          was told
    and
    peter eats meat
      was told
```

Summary

- Development of simple shell
  - Facts
  - Rules
  - Queries
- Outline of reasoning process
- Answering
  - Why questions
  - How questions

Check your understanding

- There are 6 movies stored in a system. An automatic movie playing system plays a movie to the user based on his/her mood and age.
- System should ask the user for the age and his/her mood, movie should be played accordingly.
- Define ‘movie’ clause by considering the age and the mood.
- Use write() and read() functions to ask the question and get the response from the user.
- Define ‘play’ clause which interact with user and play the required movie for the user.

Check your understanding

- movie('pasamalar', M, A) :- M = sad, A >= 50, A =< 70.
- movie('bama vijayam', M, A) :- M = happy, A >= 50, A =< 70.
- movie('thillu mullu', M, A) :- M = happy, A >= 30, A < 50.
- movie('moonram pirai', M, A) :- M = sad, A >= 30, A < 50.
- movie('nir paravai', M, A) :- M = sad, A >= 20, A < 30.
- movie('tamil padam', M, A) :- M = happy, A >= 20, A < 30.
- play(Movie) :- write('What is your age?: '), read(A), write('\nHow is your mood?: '), read(M), movie(Movie,M,A).

Check your understanding

- movie('pasamalar', M, A) :- M = sad, A >= 50, A =< 70, !.
- movie('bama vijayam', M, A) :- M = happy, A >= 50, A =< 70, !.
- movie('thillu mullu', M, A) :- M = happy, A >= 30, A < 50, !.
- movie('moonram pirai', M, A) :- M = sad, A >= 30, A < 50, !.
- movie('nir paravai', M, A) :- M = sad, A >= 20, A < 30, !.
- movie('tamil padam', M, A) :- M = happy, A >= 20, A < 30, !.
- play(Movie) :- write('What is your age?: '), read(A), write('\nHow is your mood?: '), read(M), movie(Movie,M,A).