An Implementation of Expert System For Registration of Events

Mandeep Kaur, Assistant Professor, Guru Kashi University, Talwandi Sabo, Punjab, India
Upasna Garg, Assistant Professor, Guru Kashi University, Talwandi Sabo, Punjab, India
Vinod Singla, Assistant Professor, Guru Kashi University, Talwandi Sabo, Punjab, India

ABSTRACT

A Prolog expert system (APES) supporting querying and extending the knowledge base from a command-line interface using a format oriented on natural language, with the aim of being maintainable by the domain expert (i.e. without requiring programming skills). The system will search the rules and facts in the knowledge base and request missing facts from the user. When a solution is found, the inference that lead to the conclusion (i.e. the used rules and facts) and additional solutions can be displayed. The knowledge is represented in text files containing Prolog predicates with defined operators to make it a domain-specific language which is not too far from the actual input format, making it maintainable by the domain expert without requiring Prolog or general programming skills. This leads to the core idea and goal.

Keywords & Phrases*: logic programming in Events, Events knowledge representation, expert systems, theorem proving.*

INTRODUCTION

Logic Programming is used as a declarative representation language and a model-generator is used as the problem-solver. The problem-solving task is split between the programmer, who is responsible only for ensuring the truth of programs expressed in logical form, and the theorem-prover or model-generator, which is responsible for solving problems efficiently.

- Logic programming is based on first order in predicate logic. Clauses are very common logic programming which have special forms of first order predicate logic (FOL) formula.
- Program in logic programming is a collection of clauses. Clause in logic programming adopts a special clausal notation.
- Queries are solved using resolution principle.
- Several logic programming languages have been developed since then reflecting different choices of clause selection.
-

There are three basic statements: facts, rules and queries. These are special forms of clauses as already mentioned earlier:

- Fact (Unit clause): The simplest kind of statement is called fact which states a relation between objects as such

o BROTHER (john, mike) ←

o This fact says that john is a brother of mike. BROTHER is a predicate symbol that shows relationship between two individuals named mike and john.

- **Rule (program Clause):** If we have to define a rule "X is a grandfather of y, if x is a father of z and z is a parent of y" using logic programming convention, then we write

o GRANDFATHER (x, y) ← FATHER (x, z) , PARENT (z, y)

- **Query (Goal):** Query is used to retrieve information from logic program.

o Answering a query with respect to a logic program is to determine whether the query is a logical consequence of a program or not.

o Logical consequences are obtained by applying resolution rule.

o Comma "," in the query denotes logical conjunction (Λ) to separate sub goals and is different from the comma used to separate the arguments of the predicate in the goals of the query.

o Selection of the sub goal to be reduced in a conjunctive query is arbitrary in logic programming. In any given redolent, all the sub goals must be reduced.

Prolog programs are constructed from terms:

- Constants can be either atoms or numbers:

o Atoms: Strings of characters starting with a lower-case letter or enclosed in apostrophes.

o Numbers: Strings of digits with or without a decimal point.

- Variables are strings of characters beginning with an upper-case letter or an underscore.
- Structures consist of a function or function symbol (looks like an atom), followed by a list of terms inside parentheses, separated by commas.

BACKGROUND

Prolog Logic Program:-

A clause consists of two parts: the head and the body. One side of the clause to which the arrowhead (if-then operator) points to is called the head and the other side of it is called the body. A Horn clause contains at most one literal (proposition / predicate) at the head of the clause.

Example:-

Consider the following two sentences in Predicate Logic with one head clause in each sentence.

Father (X, Y) ← Child (Y, X), Male (X).
Son (Y, X) ← Child (Y, X), Male (Y).

The above two clauses being horn clauses are constituents of a Logic Program. Let us now assume that along with the above knowledge, we have the following three data clauses:

Child (ram, dasaratha).
Male (ram).
Male (dasaratha).

Suppose that the above set of clauses is properly structured in PROLOG and the program is then compiled and executed. If we now submit the following queries (goals), the system responds correctly as follows.

1. Goal: Father (X, Y)?
Response: Father (dasaratha, ram).
2. Goal: Son (Y, X)?
Response: Son (ram, dasaratha).

But how does the system respond to our queries? We shall discuss it shortly. Before that let us learn a little bit of syntax of PROLOG programs. We take up the scene interpretation problem as an example to learn the syntax of PROLOG programming.

A. *Previous report/ Assignments:*

- **Power System Fault Diagnosis Expert System Using PROLOG**

This paper represent the diagnosis of power system fault. This expert system detects the possible faults in a power system using knowledge stored in database that can be easily changed making it flexible for different power system environments. This is a valuable system to operate in emergencies.

- **Animal expert system using Prolog**

The bird identification program has two distinct parts: the knowledge base, which contains the specific information about bird identification; and the predicates which control the user interface. By separating the two parts, a shell can be created which can be used with any other knowledge base. For example, a new expert system could be written which identified fish. It could be used with the same user interface code developed for the bird identification system.

B. Research Papers related to that field:

- Power System Fault Diagnosis Expert System Using PROLOG
By SHAHRAM B. J ADID, B.JEYASURYA, S.A.KHAPARDE
- Animal expert system using Prolog
By Steepleton

METHODOLOGY

For registration of events.

A. Given Approach:

Fuzzy Logic: A form of knowledge representation suitable for notions that cannot be defined precisely, but which depend upon their contexts.

First order predicate logic:

Analysis and appraisal of arguments. Logic is used in most intellectual activities. Studied primarily in the disciplines of philosophy, mathematics, and computer science. Logic examines general forms which arguments may take which forms are valid, and which are fallacies. It is one kind of critical thinking.

B. Your approach:

Logic Programming PROLOG:

Facts for registration of events:- The first part of the program consisted of all the facts. A event consisted of a name, place, the starting month, the starting date, the end date, create a 8 places in the database.

```
/* registerEvent(Hotel Name, Place, Start-Month, Start-Date, End-Date, Classification, Genre*/

registerEvent(a, Bradford, 9, 2, 4, adolescent, music).
registerEvent(b, Bradford, 1, 1, 1, cheap, restaurant).
registerEvent(c, Bradford, 1, 1, 1, family, music).
registerEvent(d, Leeds, 6, 4, 6, adult, cinema).
registerEvent(e, Leeds, 10, 4, 4, family, sport).
registerEvent(f, Leeds, 1, 1, 1, adult, nightclub).
registerEvent(g, Leeds, 1, 1, 1, cheap, restaurant).
registerEvent(h, Leeds, 1, 1, 1, expensive, restaurant).
foodAvailableAt(Bradford, cheap).
foodAvailableAt(Leeds, medium).
foodAvailableAt(Manchester, expensive).
foodAvailableAt(York, medium).
distance(York, Leeds, 50).
distance(York, Manchester, 100).
distance(York, Bradford, 60).
distance(Leeds, Manchester, 75).
distance(Leeds, Bradford, 25).
```

distance(Manchester, Bradford, 90).
If any person find "distance(leeds, york, X)", SWI-Prolog would answer "No".

Rules:-

```
getDates(E,M,Ds,De) :-
registerEvent(E,_,M,Ds,De,_,_).
```

This rules is relatively easy. Note that the underscore tells the Prolog system that we don't care about that information.

```
getEvent(E,P,C,G) :-
registerEvent(E,P,_,_,_,C,G).
partyEvents(E,M,D,P) :- getEvent(E,Ep,family,G),
                    near(P,Ep),
                    G \= shopping,
                    getDates(E,Em,Ds,De),
                   (allYearEvent(E) ; (Ds =< D, De >=
D, Em =:= M)).
```

The partyEvents rule is fairly complicated. Firstly, we get a family event (assuming E is passed as a variable), then we check whether it is near the place P that was passed.

Queries:

- Is there anything suitable for a party for someone whose birthday is on the 4th October in or near Leeds.
  partyEvents(X,10,4,leeds).
- Is there food available at the York, if so how expensive is it?
  foodAvailable(York, P).
- List all events in or near York.
  nearEvents(E,york).
- Get all adult events in Leeds
  getEvents(E,leeds,adult,_).

OBSERVATION

This system is coded into PROLOG and it is presently running for use validation. We can expand the system to support other diseases diagnosis within the same system shell, transforming it in register events expert system. All the facts and rules are treated as objects and stored in database to increase access time. If any person would like to set the program in its entirety, then the person can take decision from this Expert System. So, now you should be able to create relatively useful programs that can use inclusive and exclusive rules and comparisons. This expert system itself can have enough intelligence to answer network problems.

FUTURE SCOPE AND SUGGESTION:

- The knowledge based expert system with logic programming must be timely updated with newer rules about new diseases with the constant updating of the database.
- The end user will get more appropriate and accurate results.
- The interface of expert system must be user friendly .i.e. in given expert system will be used by people which can face problems in using the new system.
- The proposed expert system needs to include other diagnosis methods like new program plans.
- The development of wed based expert system poses new challenges and emphasis on more research need to be carried out.
- The expert system must be window based application as internet is not available.

REFERENCES

[1]http://dspace.library.iitb.ac.in/jspui/bitstream/10054/248/1/4471.pdf

[2] D. Karaboga, B. Basturk, A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, Journal of Global Optimization, Volume: 39, Issue: 3, pp: 459-471, Springer Netherlands, 2007. doi: 10.1007/s10898-007-9149.

[3] D. Karaboga, B. Basturk, On The Performance Of Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing, Volume 8, Issue 1, January 2008, Pages 687-697. doi:10.1016/j.asoc.2007.05.007.

[4] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, Vol: 4529/2007, pp: 789-798, Springer- Verlag, 2007, IFSA 2007. doi: 10.1007/978-3-540-72950-1_77.

[5] http://www.cse.iitd.ernet.in/~saroj/L5_Logic_prog.pdf

[6] http://www.cs.uiowa.edu/~hzhang/c123/LectureA.pdf

[7] Prof. M.S. Prasad Babu, Mrs. J. Anitha, K. Hari Krishna, "A Web Based Sweet Orange Crop Expert System using Rule Based System and Artificial Bee Colony Optimization Algorithm" , International Journal of Engineering Science and Technology ,vol.2(6),2010, pp 2408-2417

[8] M.S.Prasad Babu, J.Anitha, N.V. Ramana Murty & G. Satyanarayana", A Rule Based Medical Expert System on Dermatology", "IJAAIES "Jan-June 2010, Vol. 2(1), 2010, pp 81-86