
By Dan Rehfeldt and Magdi Ragheb


Prolog, which stands for Programming in Logic is becoming the predominant AI language in Europe and Japan. It was initially developed in 1972 by A. Colmerauer and P. Roussel at the University of Marseille, France [3]. Prolog is a simple programming language for logical programming, and as used in this work, its Explorer machine implementation provides an interface to LISP, thus generating a Knowledge Engineering Tool drawing on the strengths of both languages. Prolog enjoys an international popularity, and the publicized Japanese Fifth-Generation project has adopted Prolog as the fundamental language for the supercomputers they plan to build in the near future [4].

Like LISP [1], Prolog [3] is designed for symbolic rather than simple numerical computations. Computation in Prolog is controlled logical deductions. One states the facts known about a given situation and Prolog proceeds to tell whether or not any specific conclusion can be deduced from those facts. In the Knowledge Engineering Terminology, its control structure is logical inference. Although Prolog may be the best current implementation of logic programming, it cannot handle all the deductions that are theoretically possible in predicate calculus.

In Prolog, a programmer does not specify how the computer is to perform its assigned tasks, but just gives a description of the task as a sequence of constraints to be satisfied. This distinguishes it from the LISP programming language: in LISP one must specify the "how" of a given computation, whereas in Prolog one only specifies the "what" of a computation. In the latter case the machine is assigned the task of determining how to carry out the "how" part, which frees the programmer from worrying about the details of the algorithms which perform a given task. This is achieved at the expense of more computational resources, but those are becoming increasingly cheaper. This approach depends on assigning the different tasks of "how" and "what" to two different groups of researchers. As discussed by J. deKleer this idea may "Enable the Japanese to leap-frog over American software methodology" [5]. The combination of Prolog's theoretical elegance and its declarative interpretation will probably result in its steadily increasing acceptance worldwide. In the following, its use in building Expert Systems is demonstrated using the Animal Identification Problem [1,2], and its implementation on the Explorer machine is described. See figures 1-8, pages 13-26.

PROBLEM DESCRIPTION

For representation, we construct a logical model of the problem at hand using the technique of Fault-Tree Analysis from the field of System Analysis following the methodology described in Ref. 6. We consider a Knowledge-Base consisting of seven animals whose characteristics can be described in terms of the OR, AND and NOT logical operations:

\[
\begin{align*}
\text{Albatross} &= \text{Bird} \wedge \text{Fly} \wedge \text{Fly-Well} \\
\text{Penguin} &= \text{Bird} \wedge \text{Fly} \wedge \text{Swim} \wedge \text{Black-and-white-color} \\
\text{Ostrich} &= \text{Bird} \wedge \text{Fly} \wedge \text{Long-Neck} \\
& \wedge \text{Long-Legs} \wedge \text{Black-and-white-color} \\
\text{Giraffe} &= \text{Ungulate} \wedge \text{Long-Neck} \\
& \wedge \text{Long-Legs} \wedge \text{Dark-Spots} \\
\text{Zebra} &= \text{Ungulate} \wedge \text{Black-Stripes} \\
\text{Tiger} &= \text{Mammal} \wedge \text{Carnivore} \wedge \text{Yellow-color} \wedge \text{Black-stripes} \\
\text{Cheetah} &= \text{Mammal} \wedge \text{Carnivore} \wedge \text{Yellow-color} \wedge \text{Dark-Spots}
\end{align*}
\]

where \( (\wedge) \) denotes the AND operation and \( (\neg) \) denotes the NOT operation.

These logical relationships are shown in Figure 1. In this figure, we can notice that the parameters at the bottom of the trees in single-framed rectangles are basic facts that need to be obtained interactively from the program user, whereas those in double-framed rectangles are intermediate deductions that need to be further described. They can be logically described as follows:

\[
\begin{align*}
\text{Carnivore} &= \text{Eats-meat} \wedge (\text{Pointed-teeth} \wedge \text{Claws} \wedge \text{Forward-eyes}) \\
\text{Ungulate} &= (\text{Hoofs} \wedge \text{Mammal}) \vee \\
& (\text{Mammal} \wedge \text{Chews-Cud}) \\
\text{Mammal} &= \text{Hair} \wedge \text{Gives-milk} \\
\text{Bird} &= \text{Feathers} \wedge (\text{Fly} \wedge \text{Lay-eggs})
\end{align*}
\]

where \( (\vee) \) denotes the OR operation.

These relationships are shown in continued on page 18.
Fig. 1 Logical relationships describing animals to be identified.
Fig. I (continued) Logical Relationships describing animals to be identified.

XENIX V™
Up And Running

From operator terminals, thru multiplexing back to the computer, including Xenix™ Tape Backup, we’re up and running to help you get your special application up, running, and working for your customers. Foxfire is your fast service, no hassle, source for quality Texas Instruments computer products, as well as top brand name peripherals, and software. Plus, Foxfire has all the hardware necessary to do Xenix V™ Systems.

Call Today For More Information.

Texas Instruments

10301 Harry Hines Blvd.  foxfire, inc. Dallas, Texas 75220
Featuring Hardware From Texas Instruments  (214) 353-9543

Join the Flock . . .

of TI Specific Advertisers in
TI Professional Computing

Contact Ann Savage
(512) 250-9023
1-800-531-5093
Fig. 1 (Continued) Logical relationships describing animals to be identified.

Information Requested

Inform the TI Professional Computing Community of your users group, newsletter and/or BBS. Send complete information to: Editor, TI Professional Computing, 12416 Hymeadow Drive, Austin, TX 78750.

Samples for inclusion in Users Group Notes are welcome.

True IBM PC Compatibility

Run IBM PC software on your TI PC with the CSTI Compatibility System

- Use packages previously available only for IBM PCs
- Purchase software you need locally—no more hunting for special TI packages
- All TI version software will run as before
- IBM Graphics comes as a standard feature
- $595 complete
- CSTI consists of an add-on card and a software diskette.

IBM is a trademark of International Business Machines
TI is a trademark of Texas Instruments

Compatible Systems Corporation
Suite M1033
2111 30th Street
Boulder, Colorado 80302
(303) 444-9532
Fig. 2 Logical Descriptions of Intermediate Deductions.
Figure 2 using the AND and OR logical gates. The identification problem then becomes:

Animal = Albatross + Penguin + Ostrich + Giraffe + Zebra + Tiger + Cheetah

Substitution from Eqsns. 1 and 2 into Eqn. 3 yields the logical expression for the goal or top-event of a logical tree:

Animal = ((Feathers + (Fly * Lay-eggs))
* Black-and-white-color) + ((Hoofs *
(Hair + Gives-milk)) + ((Hair + Gives-
milk) * Chews-Cud)) * Long-Neck *
Long-Legs * Dark Spots) + ((Hoofs *
(Hair + Gives-milk)) + ((Hair + Gives-
milk) * Chews-Cud)) * Black-Stripes +
((Hair + Gives-milk) * (Eats-meal +
(Pointed-teeth * Claws * Forward-eyes))
* Yellow-color * Black-stripes) + ((Hair +
Gives-milk) * (Eats-meat + (Pointed-
teeth * Claws * Forward-eyes)) *
Yellow-color * Dark-spots)

This representation is shown in the form of a Goal Tree in Figure 3. This Goal Tree can then be translated into a Knowledge-Base and coupled to an Inference-Engine forming a Production-Rule system, as shown in Figure 4, in the form of a Prolog program.

ENHANCED PROLOG PRODUCTION-RULE SYSTEM

The prolog program is a direct translation of the logical gates representing the problem in Eqsns. 4 and Figure 3. A typical AND gate from Figure 1 is written as:

animal-is (tiger):-
identified-as (mammal),
identified-as (carnivore),
positive (characterized-by, yellow-
color),
positive (characterized-by, black-
stripes).

The (--) symbol can be considered as corresponding to (if) and the comma (,) for the (AND) logical operation.

A typical OR gate from Figure 4, is written as:

identified-as (mammal):-
positive (characterized-by, hair),
identified-as (mammal):-
positive (able-to, give-milk).

Enhancements for applications on the Explorer machine over Yazdani's version [2] are here discussed. The main enhancement to the program is accomplished by adding the statements:

abolish(pos,2),
abolish(neg,2).

Texas Instruments
TI-SERV™
Service Repair Center

- Fixed repair prices on every product we service. No hidden costs!
- Whole unit repair and refurbishment available.

DISK REPAIR
CDC 9427/9448
Century Data T25-T304
CM1 5000 and 6000 Series
Seagate 3000 Series
Seagate ST412
Rodime RC352
All "TI 5 1/4" and 8"

Mechanical repair for the following units:
Printers
CRT's
Power Supplies

CALL 713/895-4344 FOR INFORMATION AND PRICING.

Texas Instruments
Service Repair Center

- Rapid turnaround on all repairs. Five working days on TI equipment. Ten working days on third-party equipment.
- Strict testing and repair standards.
- No hidden costs!
- Over 13 years dedicated repair experience.
- All repairs backed by TI's complete Fixed Price Repair warranty.

FLOPPY REPAIR
Tandon TM100's
CDC 9409
MPI 51/52
Shugart 801
YE Data
Qume Trak 8
TEAC 55 Series
Priority services available on TI equipment repair!

PC BOARD REPAIR
TI Computers
TI Terminals
TI Printers
IBM® PC
IBM Personal Computer AT®
IBM PC/XT®

Texas Instruments
Creating useful products and services for you
which resets the memory to the original knowledge-base or long-term memory and prepares it for a new interactive inquiry by the user. The predicates (pos) and (neg) are created by the (asserta) predicate and added as new facts to the short-term memory part of the knowledge-base. A modification in the logic and the knowledge base, keeps the program from asking if the animal flies well, when it has been told that the animal doesn't even fly at all, in the case of the albatross.

The statement "asserta(X:=X)" gives a syntax error for version 1.0 of Prolog on the Explorer, so it was changed to "asserta(X)". The statement "add(X):run:-asserta(X) also gave a syntax error on the Explorer, so it was changed to "add(X):asserta(X). run:-. Since the cut operator was deleted from the assertion, it had to be added to the positive (X) and negative (X) predicates elsewhere in the modified program. Also the names under which the assertions were stored was changed to keep the program from going into an infinite loop when "not(P)" is called. The predicate "not(P)" is a good example of the classical use of the cut (!) operator:

not(P):call (P),!fail.
not(P).

The first attempt at not(P) begins to succeed if P exists. The cut operator is then used to keep Prolog from backtracking to try to satisfy "not" by other means. If call(P) succeeds, the first call to not(P) fails, and since the cut operator is used before it fails, Prolog cannot attempt the second not(P). If, however, calls(P) fails, the cut operator is not encountered, and prolog is free to attempt as many other ways as it deems necessary to satisfy not(P), namely the second call to not(P) which succeeds by its sheer simplicity.

Through the use of the (pos) and (neg) predicates, the program "remembers" any positive or negative answers about its previous inquiries, which avoids asking the same question twice. It is also capable of indentifying cases of incomplete information, where the short-term memory is cleared, and the user is directed to start a new input by initiating the search using the "start" predicate.

TYPICAL CASE RESULTS

On the Explorer machine, the program was written and edited using the Z macs editor and stored as a file with the identifier (.pl). Figure 5 shows how to call the program, by first initializing (prolog) from the Lisp listener, then using consult or reconsult to call the program file. The program is initialized by entering "start," followed by a period, as required by Prolog, and then pressing the "return" key. Figures 6, 7, and 8 shows typical case results for the identification of a bird (penguin), a mammal (giraffe), and a carnivore.

![EXPANSION PRODUCTS FOR TI PRO AND BUSINESS PRO](image)

FLOPPY DISK DRIVES
Epson SD-521 DSDD Half High Floppy Drive $119
CALENDAR CLOCK
MoynoSync dClock with SW $99
WESTERN AUTOMATION EXPANSIONS PRODUCTS
S2, 192K RAM Expansion (from 64K to 256K) $160
S1, Multifunction Card w/ 32K RAM, (expandable to 32K), serial and SASI port $355
S3, Multifunction Card w/74K RAM, (expandable to 2 Meg), serial and SASI port $695
RAM Expansion for S1 or S3 (each 64Kx8) $11
Calendar Clock Option for S1 or S3 $125

ZOBEX FIXED DISK CONTROLLER CARD (BOOTABLE)
ZX-3T Featuring emulation of Tt's Controller Card $335
RAM EXPANSION FOR TI BUS. PROD. CPU BOARD
128K $169
1 Meg $195
2 Meg $205
3 Meg $215
RAM EXPANSION CARD FOR TI PC
512K RAM Expansion (from 256K to 768K) $259
Serial Port for Above Card $85
FIXED DISK CONTROLLER FOR BUS. PRO.
Western Digital WH $379

![EPIC FIXED DISK DRIVE SUBSYSTEMS (BOOTABLE)](image)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Meg (Seagate)</td>
<td>699</td>
<td>699</td>
<td>999</td>
<td>749</td>
<td>749</td>
</tr>
<tr>
<td>10 Meg (Seagate)</td>
<td>749</td>
<td>1049</td>
<td>1249</td>
<td>999</td>
<td>999</td>
</tr>
<tr>
<td>30 Meg (Tulit)</td>
<td>949</td>
<td>1199</td>
<td>N/A</td>
<td>1249</td>
<td>1249</td>
</tr>
<tr>
<td>33 Meg (Rodrino)</td>
<td>1199</td>
<td>N/A</td>
<td>N/A</td>
<td>2249</td>
<td>2249</td>
</tr>
<tr>
<td>140 Meg (Maxtor)</td>
<td>N/A</td>
<td>N/A</td>
<td>4995</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>10 Meg Tape Backup Ext. (Irwin)</td>
<td>695</td>
<td>695</td>
<td>695</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>20 Meg Tape Backup Ext. (Irwin)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>1049</td>
<td>1049</td>
</tr>
<tr>
<td>60 Meg Tape Backup Ext. (Wangtek)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>1149</td>
<td>1149</td>
</tr>
</tbody>
</table>

*For External System add $200.00

All Epic Sale Inc. Fixed Disk Systems include: Winchester drive, controller card, cables, and installation instructions. Western Automation Systems include 81 card with 8K RAM. Systems are available without the 81 card (call for pricing). External systems include cabinet with power supply. All of the above products carry a full one year warranty and a 30 day return privilege.

TOLL FREE ORDER LINE
1-800-223-3742
TECHNICAL INFORMATION
1-214-272-7524

EPIC SALES, INC.
123 WALNUT PLANO CENTER
GARLAND, TEXAS 75042
Circle Response No. 4
Fig. 2 (Continued) Logical Descriptions of Intermediate Deductions.
THE TIPC NETWORK
We have the best prices in the Nation on:
HARD DRIVES
MEMORY EXPANSION
SOFTWARE
BUSINESS PROS
(Complete Single and Multi-user Systems)
We also offer
SERVICE AND SUPPORT
TO GO with these prices

Seagate ST255 w/Zobex Controller ............... $675
Seagate ST255 w/Xebec 1410A ............... $600
512K RAM Board (In Stock) ............... $195
Western Automation
SEEKER S-1 .................................. $315
DBASE III ................................... $495
Word Perfect Ver 4.1 .......................... $225
Business Pro 512K, 1.2 meg floppy,
color monitor, 20 meg,IBM & TI
modes, Dos 3 ......................... $5250

CALL 1-800-233-TIPC
TIPC Network
P. O. Box 2571
Sioux City, Iowa 51106
Master Card and Visa welcome

Fig. 4  Prolog-Explorer Version of the Animal Identification Problem.

Fig. 4 (Continued) Prolog-Explorer Version of the Animal Identification Problem.
Fig. 5 Program Calling from the Lisp Listener on the Explorer Machine.

Fig. 6 Typical Case for Identification of a Bird.
Fig. 7 Typical Case for Identification of a Mammal.

```
Please answer my questions with 'yes,' or 'no,' without forgetting the
period, then press the return key. Let us start.

Is the animal that you want me to identify characterized-by feathers?

[ino.]

Is the animal that you want me to identify able-to fly?

[ino.]

Is the animal that you want me to identify characterized-by hair?

[ino.]

Is the animal that you want me to identify characterized-by hoofs?

[ino.]

Is the animal that you want me to identify characterized-by long-neck?

[ino.]

Is the animal that you want me to identify characterized-by long-legs?

[ino.]

Is the animal that you want me to identify characterized-by dark-spots?

[ino.]

I can deduce from the information that you gave me that the animal you
have on your mind is the 'giraffe'.

Let us try another case by typing 'start,' without forgetting the period,
then pressing the 'return' key. Please go ahead.

[ Yes ]

Lisp Listener
```

Fig. 8 Typical Case for Identification of a Carnivore.

```
Please answer my questions with 'yes,' or 'no,' without forgetting the
period, then press the return key. Let us start.

Is the animal that you want me to identify able-to fly?

[ino.]

Is the animal that you want me to identify characterized-by hair?

[ino.]

Is the animal that you want me to identify characterized-by hoofs?

[ino.]

Is the animal that you want me to identify able-to give-milk?

[ino.]

Is the animal that you want me to identify able-to chew-cud?

[ino.]

Is the animal that you want me to identify able-to eat-meat?

[ino.]

Is the animal that you want me to identify characterized-by yellow-color?

[ino.]

Is the animal that you want me to identify characterized-by black-stripes?

[ino.]

I can deduce from the information that you gave me that the animal you
have on your mind is the 'tiger'.

Let us try another case by typing 'start,' without forgetting the period,
then pressing the 'return' key. Please go ahead.

[ Yes ]

Lisp Listener
(tiger), respectively. In all cases, an answer of "yes," or "no,", followed by a period and pressing of the "return" key is needed for a successful interactive session.

CONCLUSIONS

We demonstrated the implementation of the animal identification problem using Prolog on the Explorer Lisp machine. This is a typical application of Prolog for building Production-Rule Analysis or Expert Systems. Prolog offers here a simple way of constructing a knowledge-base coupled to an inference-engine which possesses a high degree of simplicity. The animal identification problem is generic, and can be easily applied to the identification problems arising in a variety of fields [6].

Acknowledgements

The manuscript preparation by Mrs. Catherine Pritchard is much appreciated. This work was partially supported by funds from Texas Instruments Corp., The Illinois Power Company, and the Illinois Department of Nuclear Safety.

References