SAMPLE PROGRAMS

PRINTING YOUR OUTPUT

CASE SENSITIVITY

Showing All Rules/ Facts: LISTING Command
Entering Facts: ASSERT Command
Loading Files: CONSULT Command

PROLOG FEATURES

PROLOG CONVERSATION

Comments
PROLOG Prompts
Exiting the PROLOG Interpreter
Entering the PROLOG Interpreter

INTRODUCTION

This manual was created to introduce a user to the PROLOG Interpreter software environment. It is the PROLOG Version 1.4.
Comments in PROLOG must be surrounded by:

```
/* ...
 * /  */
```

The PROLOG prompt is "continue the current command" is: 
- The PROLOG prompt to enter the next command is: 

PROLOG Prompts

```
% enu% ng
- ? [a]

You have now exited PROLOG
(CONTROL)D
Exit the PROLOG Interpreter:
```

PROLOG Interpreter

```
- ?

C-PROLOG version 1.4
```

Enter the PROLOG Interpreter:

```
enu% prolog

The following PROLOG messages will appear
```

PROLOG Interpreter

```
user inputs

The right column contains the system interaction messages and the
explanations, The right column contains the system interaction messages and the
user instructions and information. The left column contains the user instructions and
system responses shown on the terminal screen.

user inputs into the system via the keyboard

Key to the examples in this manual:

PLAIN
ITALIC
BOLD

To begin a session in the PROLOG environment, login to your SUN account and

INTRODUCTION
To begin this session, logon to your SUN account and create the text file travel as shown above.

System interaction messages and the user inputs contain the following is a sample session using the PROLOG interpreter. The left column contains the user instructions and explanations. The right column contains the interpreter's output.

The following is a sample session using the PROLOG interpreter.

Mapping of the Fact Base

```
route(s, d).        /* These are the rules */
routine(x, d).
route(s, s).

detour(montreal, ottawa).
road(albany, montreal).
road(burlington, montreal).
road(springfield, burlington).
road(montreal, springfield).
road(windsor, albany).
road(windsor, burlington).
road(montreal, windsor).

/* This is the fact base */
```

Place the following facts into the file travel:

Create a UNIX text file named: travel

PROLOG CONVERSATION
Alternative solution will be displayed. *NO*.

Type a selection and a carriage return e.g.,

```
? - road(x,montréal).

? - road(x,springfield).
```

Enter a carriage return e.g.,

```
```

If the query has two possible bindings which

A more complicated query:

```
```

Enter a carriage return e.g.,

```
```

The database solution which will make the

Query the database with variables.

```
```

The new line

statement is not complete and is continued on

Both facts were found in the database.

The comma represents the AND operation.

Query the database with 2 facts ANDed together.

```
```

Notice, this query does not contain any variables.

Check to see if there is a road between

Query the database:

```
```

This will read the rest of the facts.

Load the "travel" fact base:

```
```

Enter the PROLOG interpreter.
\[ \text{detour}(S'D') \text{ has not made any bindings.} \]

\[ \text{previously known. When NOT is applied here} \]

\[ \text{not (S=D') detour(S'D')} \]

\[ \text{not (S=D') detour(S'D')} \]

\[ \text{there are NO more solutions.} \]

\[ \text{Enter ONLY a Carriage Return <Crt>}. \]

\[ \text{The problem is fixed.} \]

\[ \text{Enter the query using the NOT operator:} \]

\[ \text{Enter ONLY a Carriage Return <Crt>}. \]

\[ \text{Notice the problem...} \]

\[ \text{designed.} \]

\[ \text{Query for two roads leading to the same} \]

\[ \text{there is no path from Springfield to Albany.} \]

\[ \text{the rule rule:} \]

\[ \text{there are no more solutions.} \]

\[ \text{Enter a Semicolon and a Carriage Return <Crt>}. \]

\[ \text{Another complicated query:} \]

\[ \text{A query with 2 variables:} \]

\[ \text{user that there are more possible bindings.} \]

\[ \text{processing will end. The YES after the} \]

\[ \text{Enter ONLY a Carriage Return <Crt>}. \]

\[ \text{The query has two possible bindings which} \]

\[ \text{The query is displayed.} \]
The "travel" database of facts will now be loaded into the PROLOG environment. The CONSULT command provides a PROLOG interpreter and load the file containing some facts and rules. The file "travel" created on page 2 is an example of a text file created.

Loading Files: CONSULT Command

Prolog Features

Your have now exited PROLOG (CONTROL-D). Exit the PROLOG interpreter.

This is not true because 5 is 2 + 3 in this case.

Enter a carriage return "c+r": Enter a carriage return "c+r":

The result is TRUE and the binding occurs.

Compare the two results:

Let the is operator:

Enter a carriage return "c+r": Enter a carriage return "c+r":

The variable X is bound to the term 5.

Enter the arithmetic function:

\[ X = X + 2 + 3. \]


**CASE SENSITIVITY**

- - route(13,-4).
- - road(3,-13).
- route(3,-4).
- route(3,-3).

Show all of the database ROUTE rules.

- ? = listing(route).

- ? = detour(albany,ottawa).
- ? = detour(montreal,ottawa).

- ? = listing(detour).

The LISTING command will print out all of the facts or rules designated.

**SHOWING ALL RULES/FACTS: LISTING Command**

The query is appended to the current fact base and can be used in the interpreter.

- ? = assert(detour(albany,ottawa)).

Enter a SINGLE fact into the database.

**ENTERING FACTS: ASSERT Command**

The ASSERT command will enter single facts into the database of facts.
Consider the following relationships:

- X and Y are first cousins
- X is a grandchild of Y
- X is a sibling of Y
- X is a grandparent of Y
- X is a sister of Y
- X is a mother
- X is a father

Consider the following relationships:

- X and Y are different
- X is the parent of Y
- X is female
- X is male
- X is the mother of Y
- X is the father of Y

Sample Programs

Print the file named "TYPESCRIPT".

```prolog
script started on

% Prolog is executed

end the prolog session

```

End the Prolog session.

```
```n
% open the scripting facility

```

Open Prolog.

Activate the scripting facility. To print the results of a session, use the SCRIPT facility available in the SUN PROLOG does not offer a printing facility for the interpreter environment. To print the results of a session, use the SCRIPT facility available in the SUN environment. The scripting facility must be activated before the work station environment. The scripting facility must be activated before the work station environment.
DATABASE FACTS

parent(julie, julia)
parent(julie, waller)
parent(waller, julie)
parent(waller, pat)
parent(louise, skipper)
parent(adolph, skipper)
parent(pat, julie)
parent(pat, seven)

female(julie)
female(sharon)
female(louise)
female(adolph)
female(pat)
female(seven)
female(waller)

male(waller)
male(seven)
male(pat)
male(adolph)
male(stanley)
male(julie)
male(sharon)
is-son(W,Y,Z,X) :- (parent(W,Z,X), parent(X,Y,Z), male(X), parent(Y,W,X)).

/* where W and Z are a mother and father, X is a son if X is a male and X has parents W and Z X is a son. */

is-mother(X) :- female(X), parent(X,Y,Z), (Y = W).  

/* X is a mother */

is-father(X) :- male(X), parent(X,Y,Z), (Y = W).  

/* X is a father */

different(X,Y) :- not(X = Y).  

/* different is true. */

/* if X and Y are not the same then the fact that they are X and Y are different */

is-mother(X,Y) :- female(X), parent(X,Y,Z), parent(Y,W,X).  

/* parent of at least 1 child. Someone is a mother if they are female and they are the X is the mother of Y */

is-father(X,Y) :- male(X), parent(X,Y,Z), parent(Y,W,X).  

/* parent of at least 1 child. Someone is a father if they are male and they are the X is the father of Y */

DATABASE RULES
grandson(X, Y) :- (parent(X,Z), parent(Z,Y), male(Z)).

/* and these parents share a parent that is the same.
   X and Y are first cousins if X and Y have separate parents
   X and Y are first cousins */

grandpa(X, Y) :- (parent(X,Z), parent(Z,Y), male(Z)).

/* X is a grandpa of Y if X is a male and Y is the parent
   of the parent of X. */

grandma(X, Y) :- (parent(X,Z), parent(Z,Y), female(Z)).

/* X is a grandparent of Y if X is a male and the parent of Y is and
   the parent of Z is X. */

sister(X, Y) :- female(Y), (parent(X,Z), parent(Z,Y), male(Z)).

/* X is a sister of Y if X is female and Y is their mother and X and
   Y share the same set of parents where Z is their father and W is their mother and X and
   Y are not the same person. */