DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

University of Chittagong

Course Code : CSE 714

Course Title : Artificial Intelligence Lab

Project Title : Food Recognizer (Expert System for Food Recognition)

Submitted by-

Joynab Bibi Nishu BInte Jamal

ID No: 17701061

Session: 2016-17

Department of Computer Science and Engineering

University of Chittagong

Table of Contents

Serial No.	Name
1	Introduction
2	Problem Statement
3	Aims
4	Objectives
5	Design
6	Core Components
7	Event List
8	Implementation
9	Dialogue Modules
10	Snapshots
11	Conclusion

Food Recognizer

1. Introduction:

Food Recognizer is an expert system . By this system anyone can find the food name. In this system, user have to interact with the system and give some inputs. The system will response according to user's input .

2. Problem Statement:

The main purpose of this project is to create a Expert System to replace the manual system for recognizing some dishes . The system will provide a comfortable environment where user can interact with the system easily . It provides instant help for recognizing food that is accurate, and error free. In the manual system , a person needs to think and decides the food name. Sometimes, a person may find it difficult to guess the food name as there are so many foods . Moreover some foods look same but taste different.

3. Aims :

The aims of this expert system are given below:

1. Revealing food name according to user's input.
2. Giving an automatic environment.
3. Giving error free output.

4. Objectives :

The objective of this expert system is to reveal food name according to user's input.

5. Design :

Design describe how the expert system react according to commands given to prolog environment for identifying the food according to user's input.

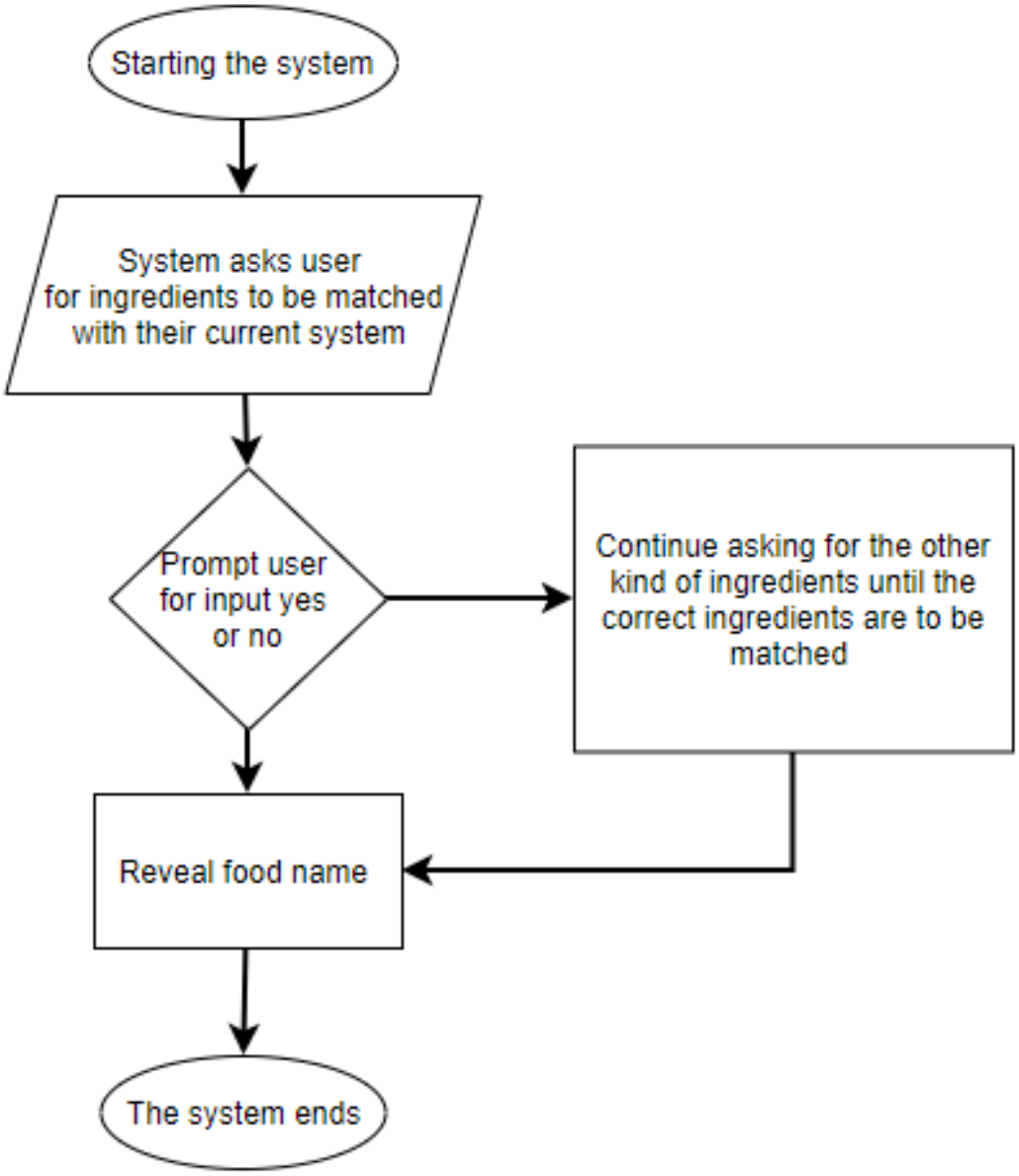


Figure 5.1: Flowchart for Food Recognizer

Here is the list of foods that the expert system can work with are-

1. Nasi lemak
2. Fried rice
3. Dandan noodle
4. Kebab
5. Biryani
6. Falafel
7. Croissant
8. Baguette
9. penne

All possible types of indications of food that we would identify are given in Table-1 . Then the system production rules are given in Table-2. Both the tables are below:

Tabel 1. Indications

No	Indications
1	Asian
2	Rice
3	Sambal
4	Coconut milk
5	Vegetables
6	Fried
7	Noodle
8	No soup
9	China origin
10	Eastern
11	Use wrap
12	Meat
13	Heavy
14	Meatball shape
15	Western
16	Pastry
17	French origin
18	Bread
19	Spicy
20	Italian

Tabel 2. System Production Rule:

Rules No.	Indications	Food Name
Rule 1	Asian, rice, sambal, coconut milk	Nasi lemak
Rule 2	Asian, vegetables, fried	Fried rice
Rule 3	Noodle, no soup, China origin	Dandan noodle
Rule 4	Eastern, meat, rice, heavy	Biryani
Rule 5	Eastern, meatball shape, vegetables	Falafel
Rule 6	Western, pastry, French	Croissant
Rule 7	Western, bread, French	Baguette
Rule 8	Western, spicy, Italian	penne

6. Core Components:

This expert system has been typically designed to provide capabilities similar to those of a human expert when performing a task. This expert system usually has two core components:

Knowledge base: This component consists of data, facts and rules for a certain topic, industry or skill, usually equivalent to that of a human expert.

Interference engine: This component uses the facts and rules in the knowledge base to find and learn new knowledge or patterns.

7. Event list:

The first matter that we should do in designing the system is making a list of every existing event.
The events are:
a) We enter any data of indication that related to the food.
b) The application does the analysis based on the user's data. The results of analysis will determine the indication list ..
c) Admin will cultivate the data on the system.
d) Admin obtains information from the data at the system.
e) The experts take over management of the system.
f) The experts get information from data management at the system.

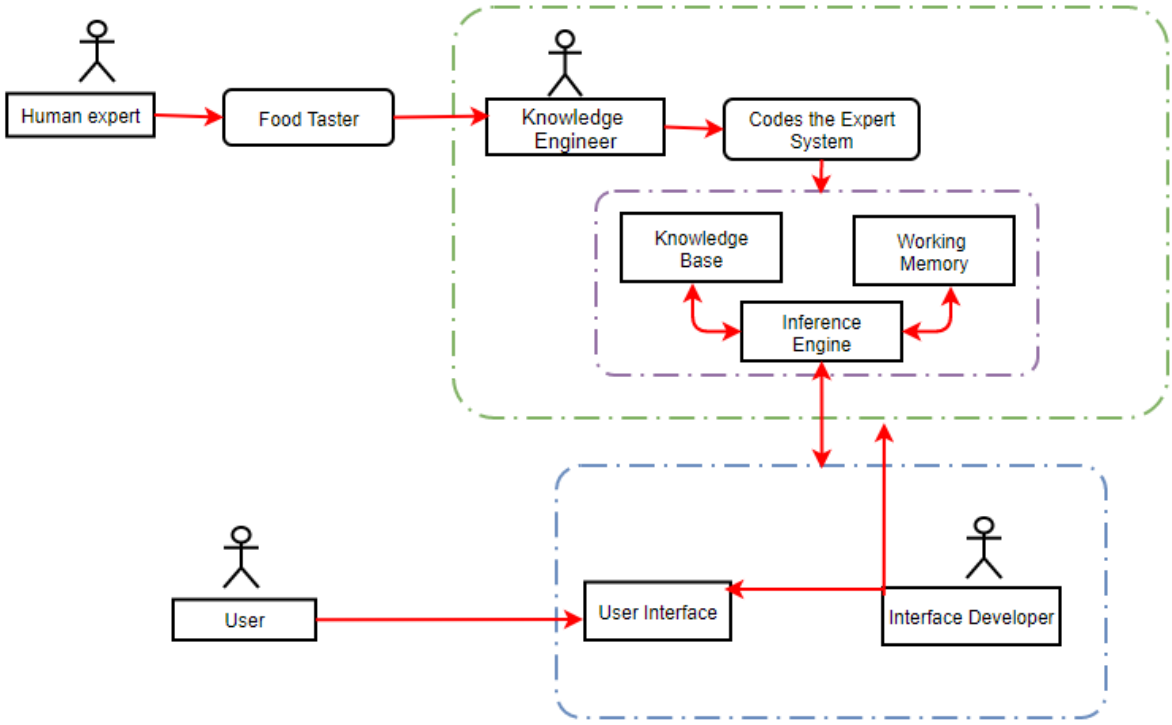


Figure: Expert System Architecture

8. Implementation:

```
domains
   food,indication = symbol
predicates
   hypothesis(food)
   verify(indication)
   response(char)
   go
clauses
   go :-
      hypothesis(food),
      write("I guess the food is ",food,"."),nl.

   go :-
      write("Sorry, I don't seem to be able to"),nl,
      write("Recognize the food"),nl.

   verify(asian) :-
      write("Is it an asian food (y/n) ?"),
      response(Reply),
```

```
    Reply='y'.

verify(rice) :-
    write("Does the food contain rice (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(sambal) :-
    write("Does the food has sambal(y/n) ?"),
    response(Reply),
    Reply='y'.

verify(coconut_milk) :-
    write("Does the food contain coconut_milk (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(veg) :-
    write("Does the food contain vegetables (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(fried) :-
    write("is the food fried (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(noodle) :-
    write("Does the food contain noodle (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(nosoup) :-
    write("Does the food contain no soup (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(chinaorigin) :-
    write("Does the food belong to China (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(eastern) :-
    write("Is it an eastern food (y/n) ?"),
    response(Reply),
    Reply='y'.
```

```
verify(usewrap) :-
    write("Does the food served with a wrap (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(meat) :-
    write("Does the food contain meat (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(heavy) :-
    write("Does the food contain heavy portion (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(meatball) :-
    write("Does the food look like meatballs (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(western) :-
    write("Does the food belong to western contries (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(pastry) :-
    write("Does the food contain pastry (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(french) :-
    write("Does the food belong to French (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(bread) :-
    write("Does the food contain bread (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(spicy) :-
    write("Is the food very spicy (y/n) ?"),
    response(Reply),
    Reply='y'.

verify(italian) :-
    write("Does the food belong to Italy (y/n) ?"),
```

```
        response(Reply),
        Reply='y'.



    hypothesis(nasi_lemak ) :-
        verify(asian),
        verify(rice),
        verify(sambal),
        verify(coconut_milk).

    hypothesis(fried_rice) :-
        verify( asian),
        verify(veg),
        verify(fried).

 hypothesis(dandan_noodle) :-
        verify(noodle),
        verify(nosoup),
        verify(chinaorigin).


 hypothesis(sichuan_noodle) :-
        verify(noodle),
        verify(chinaorigin).

hypothesis(kebab) :-
        verify(eastern),
        verify(usewrap),
        verify(meat).

hypothesis(biriyani) :-
        verify(eastern),
        verify(meat),
        verify(rice),
        verify(heavy).

hypothesis(falafel) :-
        verify(eastern),
        verify(meatball),
        verify(veg).

hypothesis(crossoint) :-
        verify(western),
        verify(pastry),
        verify(french).
```

```
hypothesis(baguette) :-
        verify(western),
        verify(bread),
        verify(french).

hypothesis(penne) :-
        verify(western),
        verify(spicy),
        verify(italian).

   response(Reply) :-
        readchar(Reply),
        write(Reply),nl.
```

9. Dialogue modules:

Some of the goal process of turbo prolog is demonstrated in Table-3 below. This is how the user will be asked a series of yes/no questions where yes means the food has those ingredients otherwise the user answers no. yes/no are the input to the system by the user.

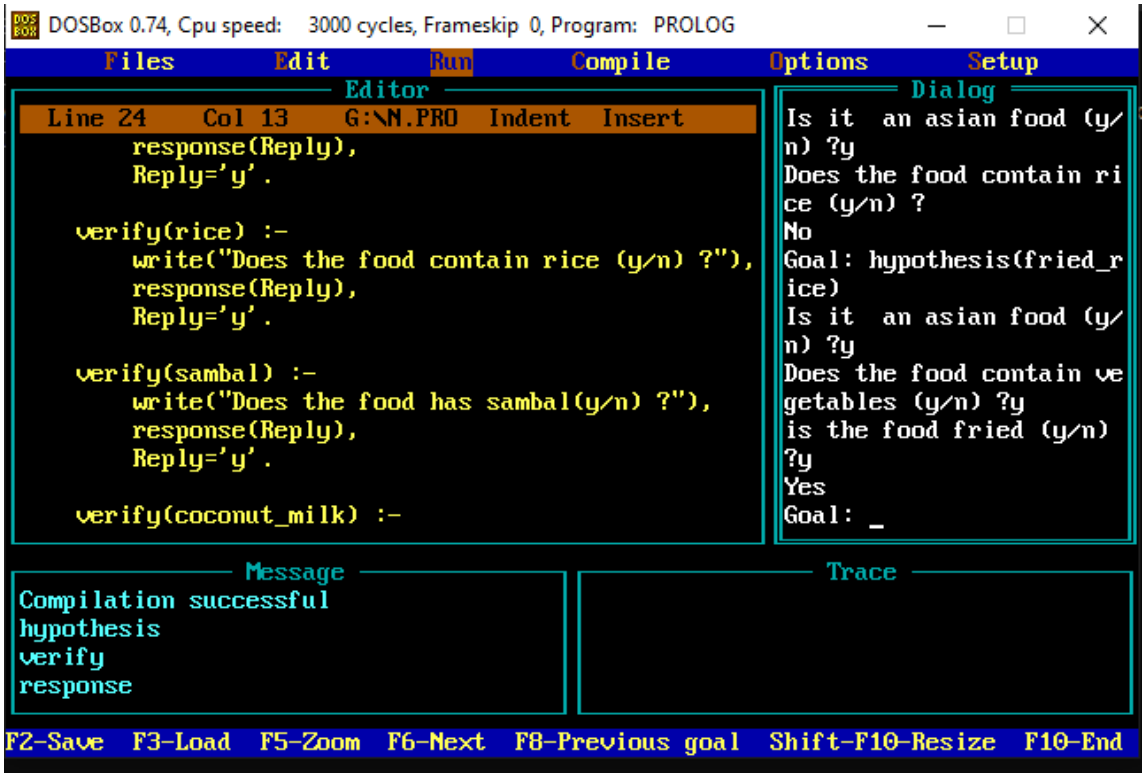
Table 3. Little demonstration of the goal process of the turbo prolog program:

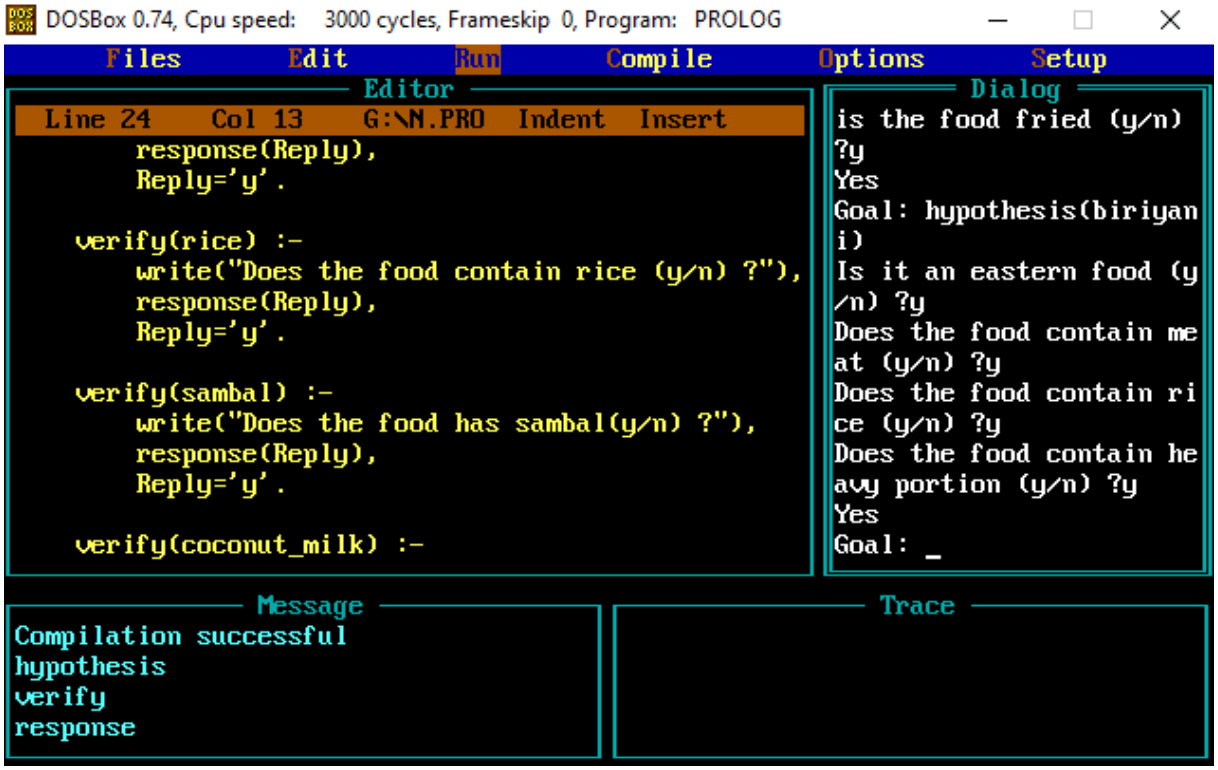
User	**Why module**
system	Is it an Asian food (y/n) ?
User	Yes
system	'Does the food contain rice (y/n) ?
User	Yes
system	Does the food has sambal(y/n) ?
User	Yes
system	Does the food contain coconut milk (y/n) ?
User	Yes
system	Does the food contain vegetables (y/n) ?
User	Yes
system	Is the food fried (y/n) ?
User	Yes
System	Does the food contain noodle (y/n) ?
User	Yes
System	Does the food contain no soup (y/n) ?
User	Yes
System	'Does the food belong to China (y/n) ?
User	Yes
System	Is it an eastern food (y/n) ?
User	Yes
System	Does the food served with a wrap (y/n) ?

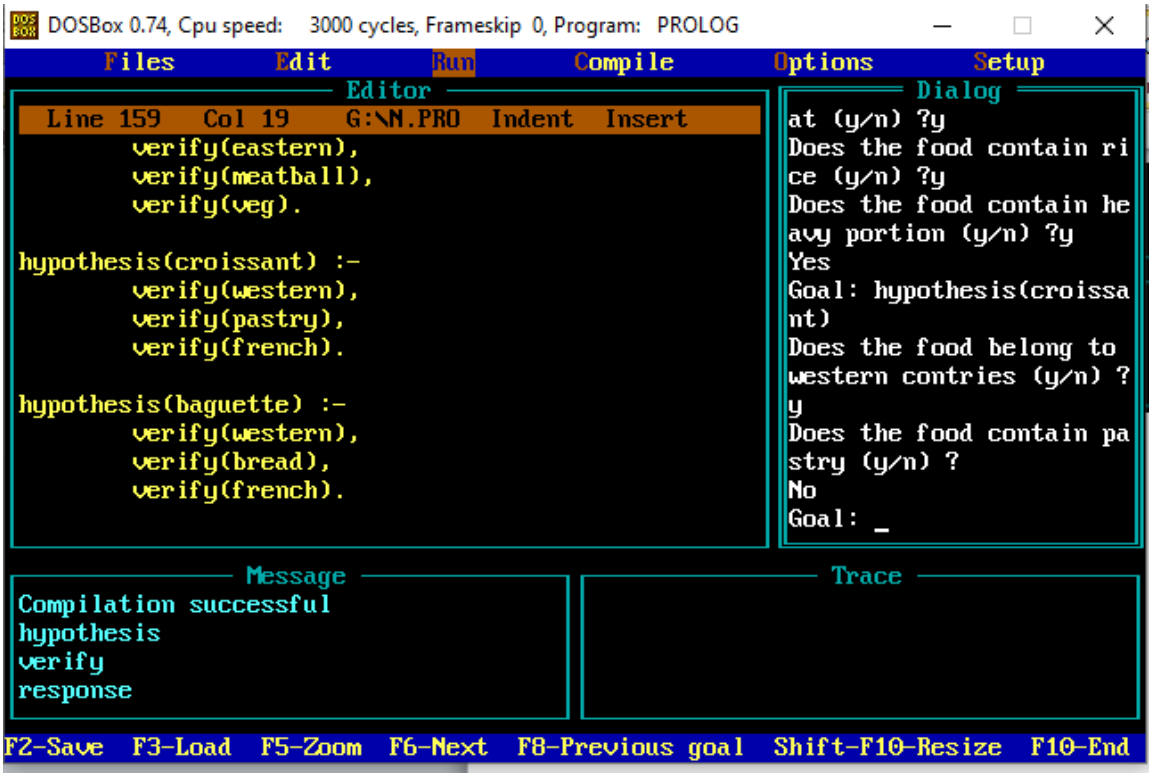
User	Yes
System	Does the food contain meat (y/n) ?
User	Yes
System	Does the food contain heavy portion (y/n) ?
User	Yes
System	Does the food look like meatballs (y/n) ?
User	Yes
System	Does the food belong to western contries (y/n) ?
User	Yes
System	Does the food contain pastry (y/n) ?
User	Yes
System	Does the food belong to French (y/n) ?
User	Yes
System	Does the food contain bread (y/n) ?
User	Yes
System	Is the food very spicy (y/n) ?
User	Yes
System	Does the food belong to Italy (y/n) ?
User	Yes

10. Snapshots:

The system was developed on prolog using DOSBOX. Some snapshots of the whole process is given below:


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: PROLOG
Files Edit Run Compile Options Setup
Editor
Line 24 Col 13 G:\N.PRO Indent Insert
response(Reply),
Reply='y'.
verify(rice) :-
write("Does the food contain rice (y/n) ?"),
response(Reply),
Reply='y'.
verify(samba1) :-
write("Does the food has samba1(y/n) ?"),
response(Reply),
Reply='y'.
verify(coconut_milk) :-
Dialog
Is it an asian food (y/n) ?y
Does the food contain rice (y/n) ?
No
Goal: hypothesis(fried_rice)
Is it an asian food (y/n) ?y
Does the food contain vegetables (y/n) ?y
is the food fried (y/n) ?y
Yes
Goal: _
Message
Compilation successful
hypothesis
verify
response
Trace
F2-Save F3-Load F5-Zoom F6-Next F8-Previous goal Shift-F10-Resize F10-End


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: PROLOG
Files Edit Run Compile Options Setup
Editor
Line 24 Col 13 G:\N.PRO Indent Insert
response(Reply),
Reply='y'.
verify(rice) :-
write("Does the food contain rice (y/n) ?"),
response(Reply),
Reply='y'.
verify(samba1) :-
write("Does the food has samba1(y/n) ?"),
response(Reply),
Reply='y'.
verify(coconut_milk) :-
Dialog
is the food fried (y/n) ?y
Yes
Goal: hypothesis(biriyani)
Is it an eastern food (y/n) ?y
Does the food contain meat (y/n) ?y
Does the food contain rice (y/n) ?y
Does the food contain heavy portion (y/n) ?y
Yes
Goal: _
Message
Compilation successful
hypothesis
verify
response
Trace



11. Conclusion:

The prolog program that I built is very simple to use and efficient. It's pretty accurate too. This system can verify the food name. This expert system has been typically designed to provide capabilities similar to those of a human expert when performing a task. The program asks a few yes/no questions to the user to note down which ingredients the user is watching. Then after knowing all the indications that the user is watching the expert system searches up on its knowledge base and then generates the most probable answer i.e. in this case the accurate food name . It is to be noted that the knowledge base is also accompanied with an interference base. Thus the expert system is more enriched and learns from experience bit by bit.