Expert System in Prolog

Department of Computer Science & Engineering

JSSATEN

by

ANURAG KAKKAR (1509110026),

ANURAG PANDEY (1509110027),

ARCHIT BHATNAGAR (1509110029),

ATIN SINGHAL (1509110035),

HIMANSHU AGRAWAL (1509110054),

HRITHIK CHAUDHARY (1509110055)

& MOHAK AHUJA (1509110066).

November 2017

Expert System in Prolog

by

Anurag Kakkar (1509110026),
Anurag Pandey (1509110027),
Archit Bhatnagar (1509110029),
Atin Singhal (1509110035),
Himanshu Agrawal (1509110054),
Hrithik Chaudhary (1509110055)
& Mohak Ahuja (1509110066).

Submitted to the Department of Computer Science
in partial fulfilment of the requirements
for the degree of
Bachelor of Technology
in
Computer Science & Engineering

JSSATEN

Dr. A.P.J Abdul Kalam Technical University, Uttar Pradesh, Lucknow.

November 2017

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:
Name : **ANURAG KAKKAR**
Roll No.: **1509110026**
Date : **17 - Nov - 2017**

Signature:
Name : **ANURAG PANDEY**
Roll No.: **1509110027**
Date : **17 - Nov - 2017**

Signature:
Name : **ARCHIT BHATNAGAR**
Roll No.: **1509110029**
Date : **17 - Nov - 2017**

Signature:
Name : **ATIN SINGHAL**
Roll No.: **1509110035**
Date : **17 - Nov - 2017**

Signature:
Name : **HIMANSHU AGRAWAL**
Roll No.: **1509110054**
Date : **17 - Nov - 2017**

Signature:
Name : **HRITHIK CHAUDHARY**
Roll No.: **1509110055**
Date : **17 - Nov - 2017**

Signature:
Name : **MOHAK AHUJA**
Roll No.: **1509110066**
Date : **17 - Nov - 2017**

CERTIFICATE

This is to certify that Project Report entitled "**Expert System in Prolog - Lung Disease Diagnosis**" which is submitted by **Anurag Kakkar, Anurag Pandey, Archit Bhatnagar, Atin Singhal, Himanshu Agrawal, Hrithik Chaudhary & Mohak Ahuja** in partial fulfilment of the requirement for the award of degree B. Tech. in Department of **Computer Science & Engineering** of **Dr. APJ Abdul Kalam Technical University**, Uttar Pradesh, Lucknow is a record of the candidate's own work carried out by him/her under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Supervisor : **Asst. Professor Vimal Gupta**

Date :

Supervisor : **Asst. Professor SS. Rawat**

Date :

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Third Year. We owe special debt of gratitude to Asst. Professor Vimal Gupta and Asst. Professor SS. Rawat, Department of Computer Science & Engineering, JSSATEN for his constant support and guidance throughout the course of our work. Their sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognisant efforts that our endeavours have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Jyoti Gautam, Head, Department of Computer Science & Engineering, JSSATEN for her full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

ABSTRACT

This project is consists of an expert system that is used to diagnose *lung disease*.

Expert systems are::

- *Decisions-making applications with high performance capabilities in comparison to the human experience especially in relation to certain problems where there is a small space for behaving.*

- *Knowledge-based systems , they are computer program s that include knowledge and analytical skills of one or more of the experts in certain field or topic .*

- *Computer applications that solve problems in an intelligent way that resembles the expert people.. Of course, this is due to the incorporation of the human experience with them.. These applications can learn and recognise the facts of the problem. They can apply this knowledge and experience in making decisions and proposing solutions. They can justify their conducts.*

TABLE OF CONTENTS

Page

INTRODUCTION

An **expert system** emulates the decision-making ability of a *human expert*.

Prolog is very well suited for *implementing* expert systems due to several reasons:

- Prolog itself can be regarded as a simple *interface engine* or theorem prover that derives conclusions from known rules. Very simple expert systems can be implemented by relying on Prolog's built-in search and backtracking mechanisms.
- Prolog data structures let us flexibly and conveniently *represent* rule-based systems that need additional functionality such as probabilistic reasoning.
- We can easily write **meta-interpreters** in Prolog to implement custom evaluation strategies of rules.

Our aim is to write an expert system that helps us *diagnose lung disease*.

Suppose, we have already obtained the following knowledge about the disease symptoms, which are rules of inference:

- If the person is suffering from *cough, shortness of breath, fever* and *shaking chills*, then the disease is **pneumonia.**
- If the person is suffering from *cough, shortness of breath, wheezing* and *chest tightening*, then the disease is **asthma.**

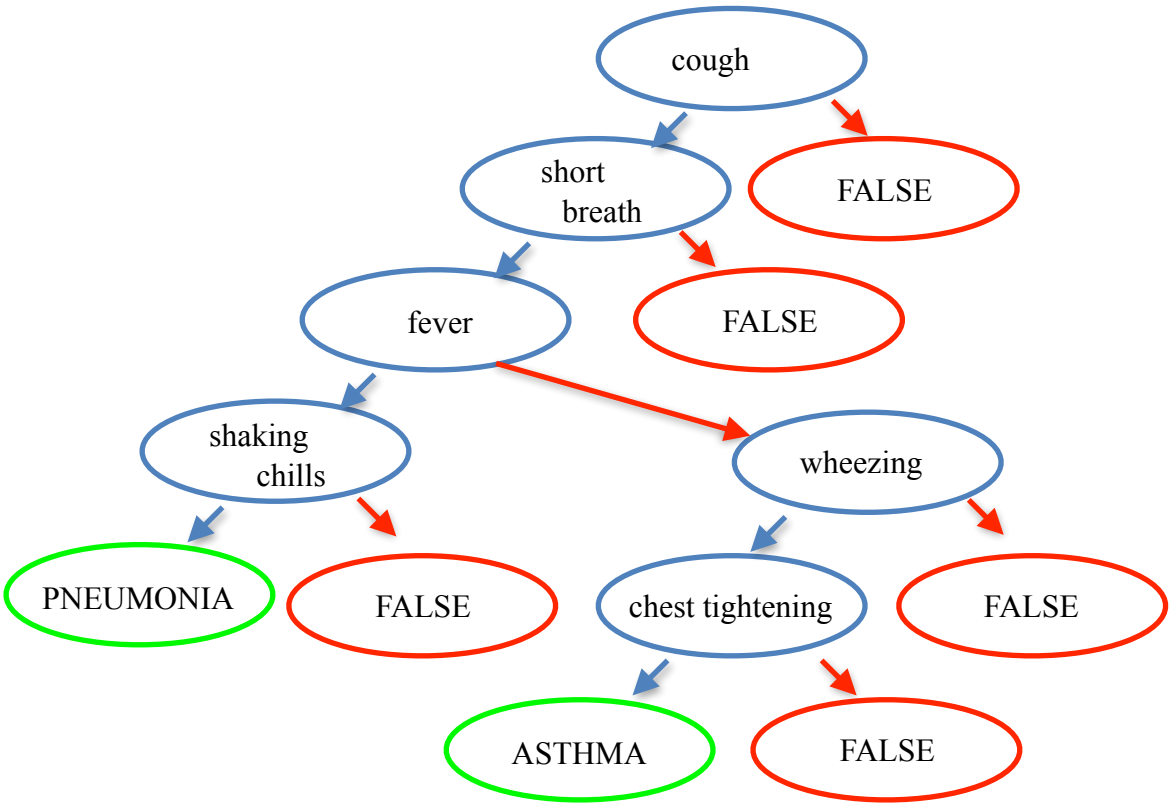
These rules are not exhaustive, but they serve as a running example to illustrate ate a few points about expert systems.

The key idea of an expert system is to derive useful new information based on user-provided input. In the following, we see several ways to do this in Prolog.

You can download a working copy of the code from https://drive.google.com/file/d/1tcgeVIFJtpMuvYRgg14kdK0DO-2VnVbe/view?usp=sharing or http://bit.ly/2hCGyZq

USING A DIFFERENT DSL

Consider now yet another way to solve the exact same problem. Let us view disease diagnose task as interpreting the following **decision diagram**, where dotted lines indicate *no*, and plain lines indicate *yes*:



SOURCE CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% lung_diagnosis.pl
%
% This is a simple program that does lung disease diagnosis.
%
% Contributors: Atin Singhal, Anurag Pandey, Hrithik Chaudhary, Archit Bhatnagar,
% Mohak Ahuja, Anurag Kakkar, Himanshu Agrawal
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


notice:-
write('This is an Expert System for Diagnosing some Lung Diseases made by students of 5th CS1 (2017-18 Batch)'),nl,
nl,
write('Contributors: Atin Singhal, Anurag Pandey, Hrithik Chaudhary, Mohak Ahuja, Anurag Kakkar, Himanshu Agrawal
'),nl,
nl,
write('To use it, just answer the quizes the systems asks you.'),nl,
nl.
```

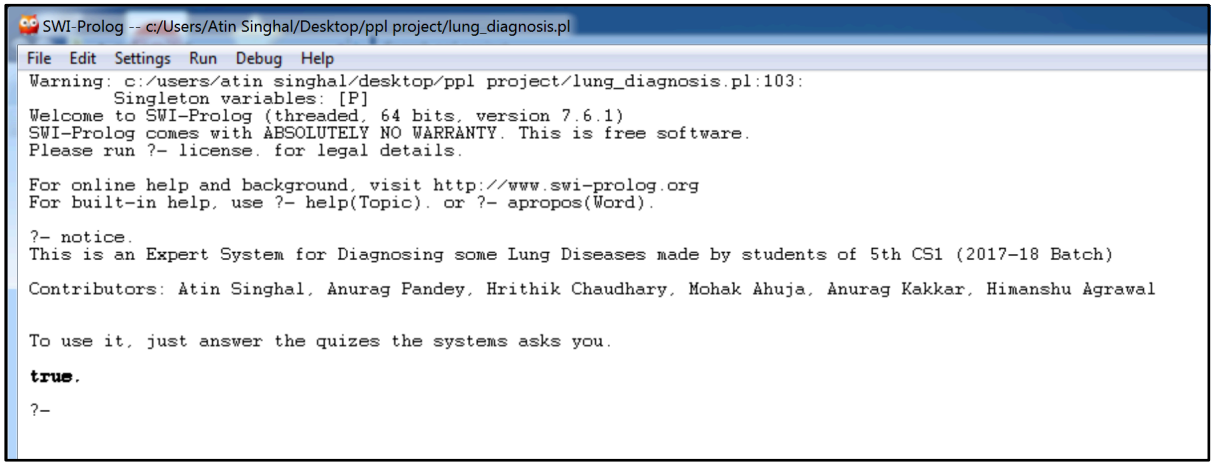
```
23
24
25  %hypothesises
26
27  disease(Patient,tuberculosis):-
28                          symptom(Patient, persistent_cough),
29                          symptom(Patient, constant_fatigue),
30                          symptom(Patient, weight_loss),
31                          symptom(Patient, lack_of_appetite),
32                          symptom(Patient, fever),
33                          symptom(Patient, coughing_blood),
34                          symptom(Patient, night_sweats).
35
36
37  disease(Patient,pneumonia):-
38                          symptom(Patient, cough),
39                          symptom(Patient, fever),
40                          symptom(Patient, shaking_chills),
41                          symptom(Patient, shortness_of_breath).
42
43
44  disease(Patient,byssinosis):-
45                          symptom(Patient, chest_tightness),
46                          symptom(Patient, cough),
47                          symptom(Patient, wheezing).
48
49
50  disease(Patient,pertusis):-
51                          symptom(Patient, runny_nose),
52                          symptom(Patient, mild_fever).
53
54
```

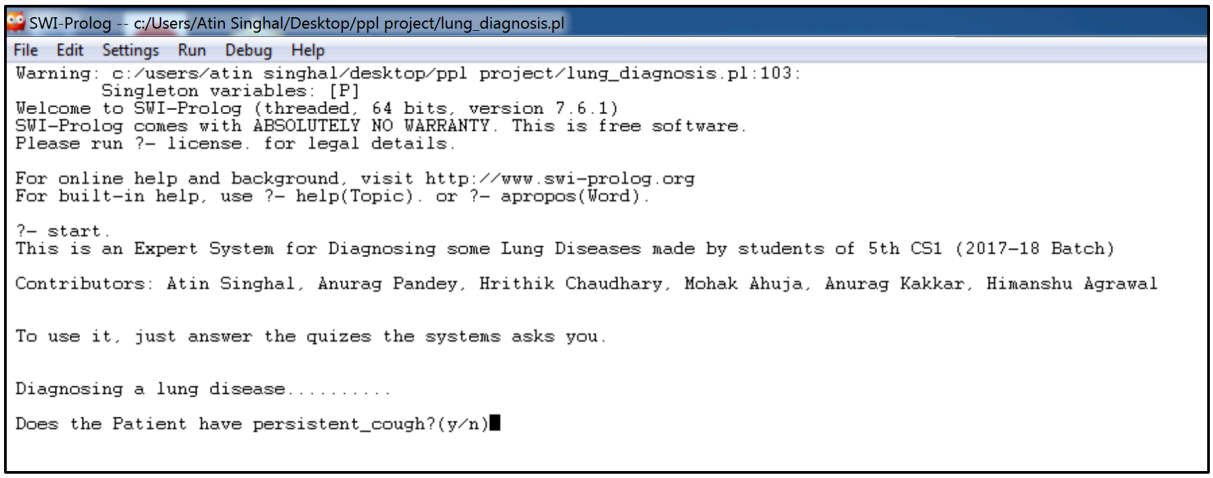
```
55 disease(Patient,pneumoconiosis):-
56                         symptom(Patient,chronic_cough),
57                         symptom(Patient,shortness_of_breath).
58
59
60 disease(Patient,sarcoidosis):-
61                         symptom(Patient, dry_cough),
62                         symptom(Patient, shortness_of_breath),
63                         symptom(Patient, mild_chest_pain),
64                         symptom(Patient, scaly_rash),
65                         symptom(Patient, fever),
66                         symptom(Patient, red_bumps_on_legs),
67                         symptom(Patient, sore_eyes),
68                         symptom(Patient, swollen_ankles).
69
70
71 disease(Patient,asbestosis):-
72                         symptom(Patient, chest_tightness),
73                         symptom(Patient, shortness_of_breath),
74                         symptom(Patient, chest_pain),
75                         symptom(Patient, lack_of_appetite).
76
77
78 disease(Patient,asthma):-
79                         symptom(Patient, wheezing),
80                         symptom(Patient, cough),
81                         symptom(Patient, chest_tightness),
82                         symptom(Patient, shortness_of_breath).
83
84
85 disease(Patient,bronchiolitis):-
86                         symptom(Patient, wheezing),
```

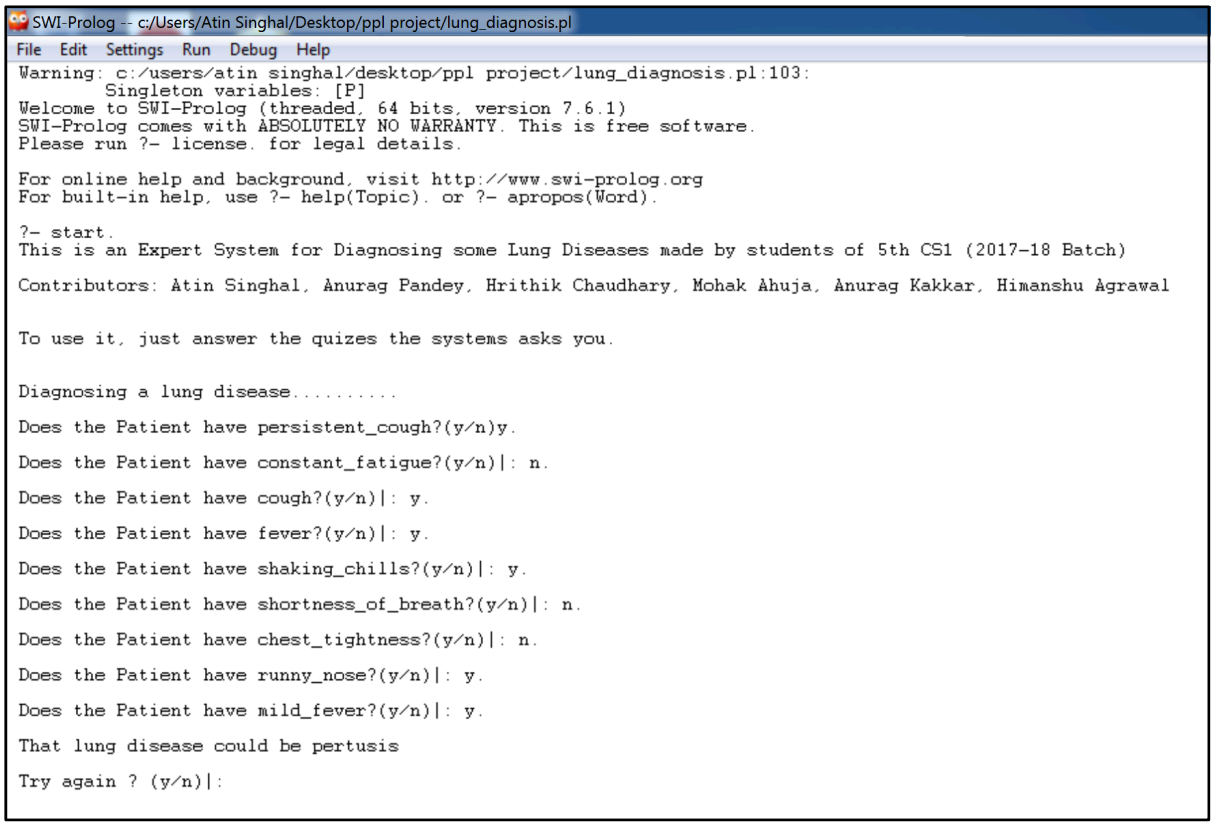
```
87                          symptom(Patient, fever),
88                          symptom(Patient, blue_skin),
89                          symptom(Patient, rapid_breath).
90
91
92  disease(Patient,influenza):-
93                          symptom(Patient, headache),
94                          symptom(Patient, fever),
95                          symptom(Patient, shaking_chills),
96                          symptom(Patient, nasal_congestion),
97                          symptom(Patient, runny_nose),
98                          symptom(Patient, sore_throat).
99
100
101 disease(Patient,lung_cancer):-
102                         symptom(Patient, cough),
103                         symptom(Patient, fever),
104                         symptom(Patient, hoarseness),
105                         symptom(Patient, chest_pain),
106                         symptom(Patient, wheezing),
107                         symptom(Patient, weight_loss),
108                         symptom(Patient, lack_of_appetite),
109                         symptom(Patient, coughing_blood),
110                         symptom(Patient, headache),
111                         symptom(Patient, shortness_of_breath).
112
113
114 /*Ask rules*/
115
116 symptom(P, Val):-ask('Does the Patient have',Val).
117 ask(Obj, Val):-known(Obj, Val, true),!.
118 ask(Obj, Val):-known(Obj, Val, false),!, fail.
```

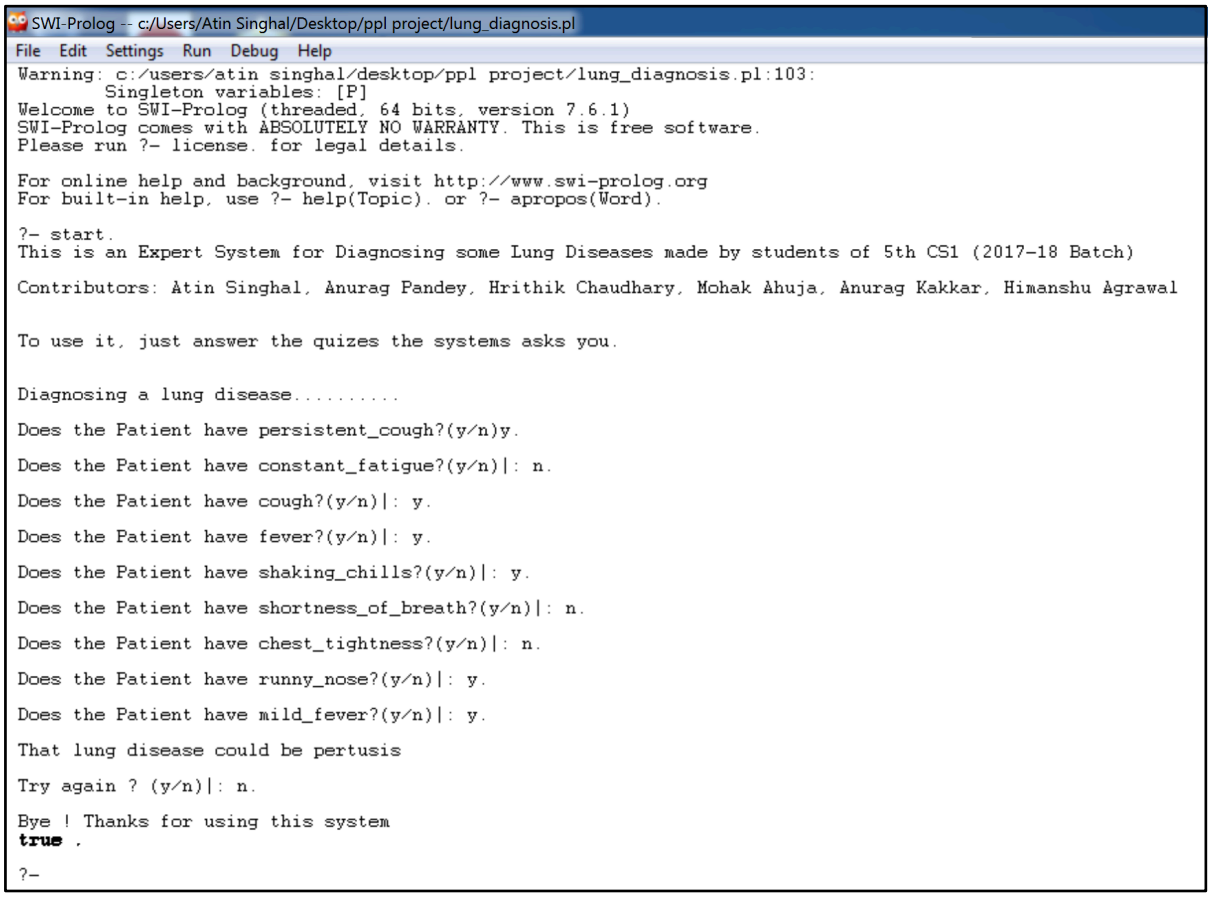
```
119 ask(Obj, Val):-nl,write(Obj),write(' '),
120             write( Val) , write('?(y/n)'), read(Ans), !,
121             ((Ans=y, assert(known(Obj, Val, true)));
    (assert(known(Obj, Val, false)),fail)).
122
123
124 diagnose:-nl,write('Diagnosing a lung
    disease.........'),nl,disease(symptom,Disease) ,!,nl,
125             write('That lung disease could be '),
    write(Disease).
126
127 diagnose:- nl, write('Sorry,we may not be able to diagnose the
    desease!!').
128
129 start:-notice,repeat, abolish(known/3),dynamic(known/3),
    retractall(known/3), diagnose,nl,nl, write('Try again ? (y/
    n)'),read(Resp),\+ Resp=y,
130         nl,write('Bye ! Thanks for using this
    system'),abolish(known,3) .
```

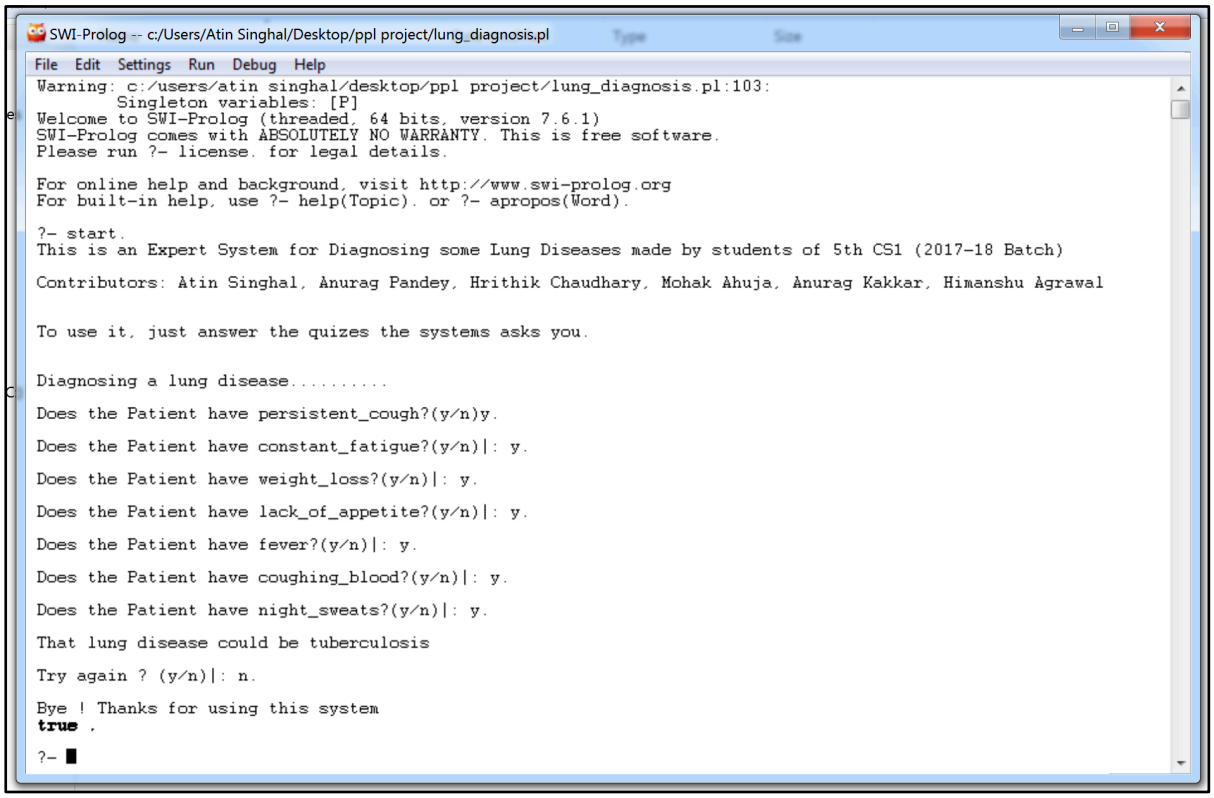
OUTPUTS

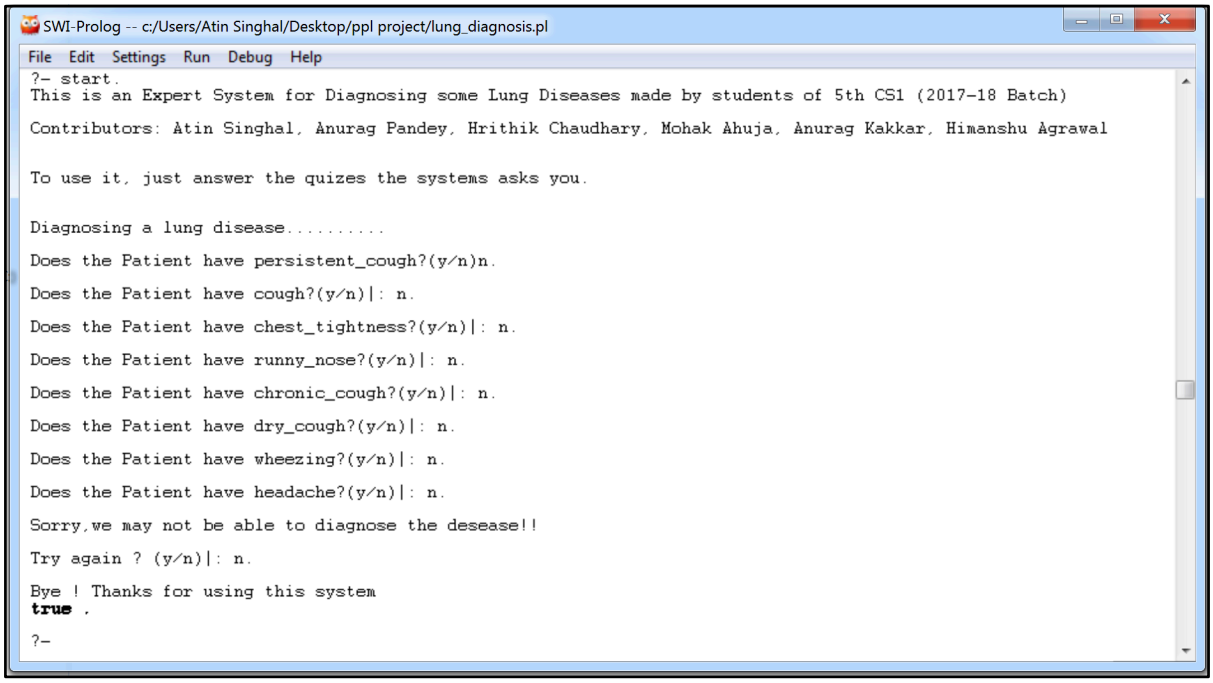

SWI-Prolog -- c:/Users/Atin Singhal/Desktop/ppl project/lung_diagnosis.pl
File Edit Settings Run Debug Help
Warning: c:/users/atin singhal/desktop/ppl project/lung_diagnosis.pl:103:
Singleton variables: [P]
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- notice.
This is an Expert System for Diagnosing some Lung Diseases made by students of 5th CS1 (2017-18 Batch)
Contributors: Atin Singhal, Anurag Pandey, Hrithik Chaudhary, Mohak Ahuja, Anurag Kakkar, Himanshu Agrawal
To use it, just answer the quizes the systems asks you.
true.
?-


SWI-Prolog -- c:/Users/Atin Singhal/Desktop/ppl project/lung_diagnosis.pl
File Edit Settings Run Debug Help
Warning: c:/users/atin singhal/desktop/ppl project/lung_diagnosis.pl:103:
Singleton variables: [P]
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- start.
This is an Expert System for Diagnosing some Lung Diseases made by students of 5th CS1 (2017-18 Batch)
Contributors: Atin Singhal, Anurag Pandey, Hrithik Chaudhary, Mohak Ahuja, Anurag Kakkar, Himanshu Agrawal
To use it, just answer the quizes the systems asks you.
Diagnosing a lung disease.....
Does the Patient have persistent_cough?(y/n)


SWI-Prolog -- c:/Users/Atin Singhal/Desktop/ppl project/lung_diagnosis.pl
File Edit Settings Run Debug Help
Warning: c:/users/atin singhal/desktop/ppl project/lung_diagnosis.pl:103:
Singleton variables: [P]
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- start.
This is an Expert System for Diagnosing some Lung Diseases made by students of 5th CS1 (2017-18 Batch)
Contributors: Atin Singhal, Anurag Pandey, Hrithik Chaudhary, Mohak Ahuja, Anurag Kakkar, Himanshu Agrawal
To use it, just answer the quizes the systems asks you.
Diagnosing a lung disease.....
Does the Patient have persistent_cough?(y/n)y.
Does the Patient have constant_fatigue?(y/n)|: n.
Does the Patient have cough?(y/n)|: y.
Does the Patient have fever?(y/n)|: y.
Does the Patient have shaking_chills?(y/n)|: y.
Does the Patient have shortness_of_breath?(y/n)|: n.
Does the Patient have chest_tightness?(y/n)|: n.
Does the Patient have runny_nose?(y/n)|: y.
Does the Patient have mild_fever?(y/n)|: y.
That lung disease could be pertusis
Try again ? (y/n)|:


SWI-Prolog -- c:/Users/Atin Singhal/Desktop/ppl project/lung_diagnosis.pl
File Edit Settings Run Debug Help
Warning: c:/users/atin singhal/desktop/ppl project/lung_diagnosis.pl:103:
Singleton variables: [P]
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- start.
This is an Expert System for Diagnosing some Lung Diseases made by students of 5th CS1 (2017-18 Batch)
Contributors: Atin Singhal, Anurag Pandey, Hrithik Chaudhary, Mohak Ahuja, Anurag Kakkar, Himanshu Agrawal
To use it, just answer the quizes the systems asks you.
Diagnosing a lung disease.....
Does the Patient have persistent_cough?(y/n)y.
Does the Patient have constant_fatigue?(y/n)|: n.
Does the Patient have cough?(y/n)|: y.
Does the Patient have fever?(y/n)|: y.
Does the Patient have shaking_chills?(y/n)|: y.
Does the Patient have shortness_of_breath?(y/n)|: n.
Does the Patient have chest_tightness?(y/n)|: n.
Does the Patient have runny_nose?(y/n)|: y.
Does the Patient have mild_fever?(y/n)|: y.
That lung disease could be pertusis
Try again ? (y/n)|: n.
Bye ! Thanks for using this system
true .
?-


SWI-Prolog -- c:/Users/Atin Singhal/Desktop/ppl project/lung_diagnosis.pl
File Edit Settings Run Debug Help
Warning: c:/users/atin singhal/desktop/ppl project/lung_diagnosis.pl:103:
Singleton variables: [P]
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.1)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- start.
This is an Expert System for Diagnosing some Lung Diseases made by students of 5th CS1 (2017-18 Batch)
Contributors: Atin Singhal, Anurag Pandey, Hrithik Chaudhary, Mohak Ahuja, Anurag Kakkar, Himanshu Agrawal
To use it, just answer the quizes the systems asks you.
Diagnosing a lung disease.....
Does the Patient have persistent_cough?(y/n)y.
Does the Patient have constant_fatigue?(y/n)|: y.
Does the Patient have weight_loss?(y/n)|: y.
Does the Patient have lack_of_appetite?(y/n)|: y.
Does the Patient have fever?(y/n)|: y.
Does the Patient have coughing_blood?(y/n)|: y.
Does the Patient have night_sweats?(y/n)|: y.
That lung disease could be tuberculosis
Try again ? (y/n)|: n.
Bye ! Thanks for using this system
true .
?-


SWI-Prolog -- c:/Users/Atin Singhal/Desktop/ppl project/lung_diagnosis.pl
File Edit Settings Run Debug Help
?- start.
This is an Expert System for Diagnosing some Lung Diseases made by students of 5th CS1 (2017-18 Batch)
Contributors: Atin Singhal, Anurag Pandey, Hrithik Chaudhary, Mohak Ahuja, Anurag Kakkar, Himanshu Agrawal
To use it, just answer the quizes the systems asks you.
Diagnosing a lung disease.....
Does the Patient have persistent_cough?(y/n)n.
Does the Patient have cough?(y/n)|: n.
Does the Patient have chest_tightness?(y/n)|: n.
Does the Patient have runny_nose?(y/n)|: n.
Does the Patient have chronic_cough?(y/n)|: n.
Does the Patient have dry_cough?(y/n)|: n.
Does the Patient have wheezing?(y/n)|: n.
Does the Patient have headache?(y/n)|: n.
Sorry,we may not be able to diagnose the desease!!
Try again ? (y/n)|: n.
Bye ! Thanks for using this system
true .
?-

FUTURE ENHANCEMENTS

Knowledge Base

The first component, the knowledge base, is a collection of facts, rules and procedures organised into schemas. It is the assembly of all information and knowledge specific to the area of accounting in which the expert system is applied.

Expanding the knowledge base of the Expert system to various other diseases related to other parts of the Human Body.

Inference Engine

The second component, the inference engine, performs the reasoning function in the expert system. The inference engine is the "brain" of the expert system and is the component that processes information. The engine recognises the most appropriate rule from the knowledge base and matches it with information supplied by the user. The inference engine links the rules in the knowledge base to derive a conclusion and communicates this conclusion to the user through the interface.

By expanding the Knowledge base of the Expert System, we will improve the accuracy of the same.

User Interface

The third component, the user interface, controls user interaction, accepting commands from the computer keyboard and displaying the results generated by the inference engine.

In the future build, we will improve the user experience.

CONCLUSION

We have now seen three different ways to implement an expert system in Prolog:

- Direct Prolog Implementation
- Devising and Interpreting a Domain-Specific Language
- Using a completely different Domain-Specific Language.

Each of these approaches was rather easy to implement in Prolog, and there are several other DSLs that would also be suitable. The question thus arises: Which DSL, if any, should we choose to implement expert systems in Prolog? Let us briefly consider the main points we have seen:

1. Using Prolog *directly* is straight-forward. However, a naive implementation has a few drawbacks. In our case, the same question was necessarily asked repeatedly.
2. Using a *domain-specific language* lets us cleanly separate the main logic of the expert system from additional features, such as keeping track of already answered questions.
3. A DSL based on decision diagram diagrams is very easy to interpret and automatically avoids redundant questions.

From these points alone, option (3) seems very attractive. However, it also raises a few important questions: First, how was the decision diagram even *obtained* and does it faithfully model the conditions e want to express? It is rather easy to do it by hand in this example, but how would you do it in more complex cases? Second, how *costly* is the transformation from a rather straight-forward fact base as in option (2) to using decision diagrams instead? Third, is this really a good diagram, and what do we mean by *good*? Are there orderings of nodes that let us *reduce* the number of questions? In the worst case, on average, in the best case? Fourth, how extensible is the language of decision diagrams? For example, can *all* diseases diagnosis tasks be modelled in this way? etc.

These questions show that the best choice depends on many factors.

References

- Book: Programming in Prolog, by W.F. Clocksin and C.S. Mellish
- A tutorial on Prolog at https://www.cpp.edu/~jrfisher/www/prolog_tutorial/pt_framer.html
- A tutorial and introduction of Prolog at https://www.tutorialspoint.com/introduction_to_prolog/index.asp
- A freeway version of Prolog from http://www.swi-prolog.org/Download.html
- Quick start guide of SWI-Prolog from http://www.swi-prolog.org/pldoc/man?section=quickstart
- Warning Signs of Lung diseases from http://www.lung.org/lung-health-and-diseases/warning-signs-of-lung-disease/
- Overview of Lung diseases from https://www.webmd.com/lung/lung-diseases-overview#1
- Overview of Lung diseases from https://www.webmd.com/lung/lung-diseases-overview#2