**CS571 Expert System for Diagnosis of Illness.**

**Akhil Sriram.**

**srira002@cougars.csusm.edu**

**State of my system – Completed.**

**The Checklist**

· ✓**\_\_\_ Did you follow all the requirements in implementing the system?**

· ✓**\_\_\_ Did you create one Word file of your report with the cover page and section headers as specified?**

· ✓**\_\_\_ Did you answer all questions per section?**

==============================================================

**Section 1: The purpose of the Expert System**

The knowledge based expert system which is built by me falls under the domain of medical diagnosis of many illnesses like cold, typhoid, malaria, flu, tonsillitis, measles, asthma, chickenpox, insomnia, kidney infection, sinusitis, indigestion, mouth ulcer, diabetes and corona and to diagnose these illnesses we use the symptoms of those diseases.

The system reaches at the conclusion of the correct illness by proposing the exact illness which would have affected the user with some confidence factor, gives prescription and advice along with wishes of recovery based on the symptoms which are typed or entered into the expert system by its user.

As mentioned above the expert system identifies the illness by reading the symptoms given by the user, so here the user needs to give inputs that whether he has the symptom or not.

So the following must be given as the input by the users whenever the system asks the user

Does the patient have the following symptoms?

a. The user must type “y.” or “yes.” if he/she has the symptom..

b. The user must type “n.” or “no.” if he/she does not have the symptom.

Some of the possible conclusions given to the user when the user responses are entered are shown below (I did not list all the possible conclusions because that would make the report too long)

For Cold:

|  |
| --- |
| As per the symptoms it seems like you have cold  with confidence factor 90  Prescription and Advice:  1: Cheston/tab  2: Acetaminophen/tab  3: Aleve/tab  4: Phenylephrine nasal spray  Please drink warm soup  hope you get well soon,Take Care |

For typhoid

|  |
| --- |
| As per the symptoms it seems like you have typhoid  with confidence factor 95  Prescription and Advice:  1: Ceftriaxone/tab  2: Ampicillin/tab  3: Zithromax/tab  4: ciprofloxacin/tab  Please take complete bed rest and take soft Diet and avoid outside food.  hope you get well soon,Take Care |

For Malaria:

|  |
| --- |
| As per the symptoms it seems like you have malaria  with confidence factor 80  Prescription and Advice:  1: Aralen/tab  2: Qualaquin/tab  3: Plaquenil/tab  4: Mefloquine  Wear clothes that cover most of the body,use mosquito repellent lotions  hope you get well soon,Take Care |

For Insomnia

|  |
| --- |
| As per the symptoms it seems like you have insomnia  with confidence factor 85  Prescription and Advice:  1: Zaleplon/tab  2: Doxepin/tab  3: Rozerem/tab  4: Restoril/tab  Donot consume caffeine ,alcohol,take regular exercise and follow timings for sleeping  hope you get well soon,Take Care |

For Corona

|  |
| --- |
| As per the symptoms it seems like you have corona  with confidence factor 90  Prescription and Advice:  1: Currently no medication is recommended but plasma treatment and hydroxychloroquine are being used in some countries  Wash hands often, use sanitizer, wear mask when you go out don't touch your eyes,nose and mouth..STAY HOME STAY SAFE..Social Distancing is the key  hope you get well soon,Take Care |

FWC:

In Forward chaining first the available facts are considered and attempts are made to draw the goals . It works for the problems that start from data collection.

example : monitoring,planning etc .Therefore here in FWC forming a goal is difficult

BWC:

In backward chaining the hypothesis starts from analyzing what the goal is,and then looks for the evidence to prove these assumptions or the hypothesis.So in BWC what we do is we begin with the assumptions/hypothesis and then gather information till we prove the assumption true or false. Here the system acquires the data through inputs interactively i.e by asking the user to enter the valid user inputs.BWC is very effective and easy in forming a goal.The Prolog Interpreter does a Depth First Search for BWC.

So in my domain which is for diagnosis of illness BWC is suitable because we start from hypothesis/assumption that the user has some illness and then try to prove it by collecting user inputs i.e we check the facts which are true or false and whenever the facts are false we backtrack to the previous node in the DFS tree and then try to verify other facts which can be used to prove rules of illness and then this can be used to prove our hypothesis/assumption. For example if we try to check symptoms if all the check on the right hand side are verified except one then it returns the left hand side as false but it would backtrack to next node and it would now does not check for the symptoms which are verified and also does not check symptoms which are not verified.Hence BWC is useful in this domain.

If we try to use FWC we have to keep a large number of facts stored and what if the fact would not be true .i.e. is false then we cant drive to the goal quickly i.e. can't diagnose the illness quickly whereas if we use BWC we can prove our assumption quickly .i.e. We can diagnose the illness by collecting user input by interacting with the user and if some fact is not true then we can easily backtrack i.e. a lot of combinations of symptoms can be used and still the goal can be proven/disproven quickly.

**Section 2: Expert System Implementation**

Facts: We know that facts are some predicate expressions .i.e. predicates followed by some arguments which can be used to make declarative statement about the problem

In my expert system which is used to diagnose illness we use facts to make our system functional i.e.we use facts to draw some declarative information. Some of the facts which my expert system uses are

check(Fact)

|  |
| --- |
| check(headache)  check(runny\_nose)  check(sneezing)  check(sore\_throat)  check(nasal\_congestion) |

**The types of Rules**

Rule is an add-on of the fact but the only difference is rule has conditions that must be satisfied for the latter to be true. Rule has two parts where the first part is the same as the fact that has the predicate and the argument. Whereas the second part consists of clauses that might be either a fact or rule. We can say that the rule is proved as true if the clause is proved or positive. The two parts of the rule are separated by “ :- “ mark which is similar to “if” in other languages.

In simple words rules are used to infer facts from facts. The types of rules which are used in my expert system are:

The assumption rule

These rules are to find out illness with certainty factor

assumption((Illness),CF)

The assumption rule has two parameters/arguments i.e illness and CF(confidence factor) which are facts themselves forms the left hand side of the rule and this left side of the rule is true if right side of the rule is true i.e if the symptoms are verified and vice versa.

The assumption rule is used to identify the illness with some confidence factor.

An example of one such rule is

|  |
| --- |
| assumption((cold),90) :-  check(headache),  check(runny\_nose),  check(sneezing),  check(sore\_throat),  check(nasal\_congestion). |

Here if the right hand side is verified .i.e. check(headache), check(runny\_nose), check(sneezing), check(sore\_throat), check(nasal\_congestion). then the left hand is true ie assumption((cold),90)

Here the cold which is a fact is true only when all the facts on the right hand side are true even if one is not true then it is false.

The medication rule

These rules are used to provide medication to the illness

medication((Illness),CF),

The medication rule has two parameters/arguments i.e illness and CF(confidence factor) which is the left hand side of the rule and this left side of the rule is true if right side of the rule is true i.e if the symptoms are verified and vice versa.

An example of one such rule is

|  |
| --- |
| medication(typhoid,95) :- mtyphoid, !. |

Here if the right hand side is verified .i.e. mthyhoid, !. then the left hand side is true i.e. medication(typhoid,95) :- mtyphoid, !. is true

But the only difference here when compared to assumption rule is that here the right hand side is a sub- goal rule which writes some prescription and advice

|  |
| --- |
| mtyphoid :-  write('Prescription and Advice:'),  nl,  write('1: Ceftriaxone/tab'),  nl,  write('2: Ampicillin/tab'),  nl,  write('3: Zithromax/tab'),  nl,  write('4: ciprofloxacin/tab'),  nl,  write('Please take complete bed rest and take soft Diet and avoid outside food'),  nl. |

We can also include another subgoal rule

|  |
| --- |
| medication(cold,90) :- mcold, !. |

Here if the right hand side is verified .i.e. mcold, !. then the left hand side is true i.e medication(cold,90) :- mcold, !.

But the only difference here when compared to assumption rule is that here the right hand side is a sub- goal rule i.e. mcold is a subgoal rule which again has some subgoal rule.

|  |
| --- |
| mcold :- mcoldprescription, !. |

And then the mcoldprescription is used to give prescriptions which are written statements.

|  |
| --- |
| mcoldprescription:-  write('Prescription and Advice:'),  nl,  write('1: Cheston/tab'),  nl,  write('2: Acetaminophen/tab'),  nl,  write('3: Aleve/tab'),  nl,  write('4: Phenylephrine nasal spray'),  nl,  write('Please drink warm soup'),  nl. |

Similarly we have different subgoals too.

Also we have other types of rules like

|  |
| --- |
| ask(Question) :-  write('Does the patient have following symptom:'),  write(Question),  write('? '),  read(Response),  nl,  ( (Response == yes ; Response == y)  ->  asserta(yes(Question)) ;  asserta(no(Question)), fail). |

This above rule is used to ask a question which takes input from the user as the response and uses asserta clauses to ascertain that the asked question is provided the correct response accordingly i.e. this rule is true only if the entire facts on the right side are true.

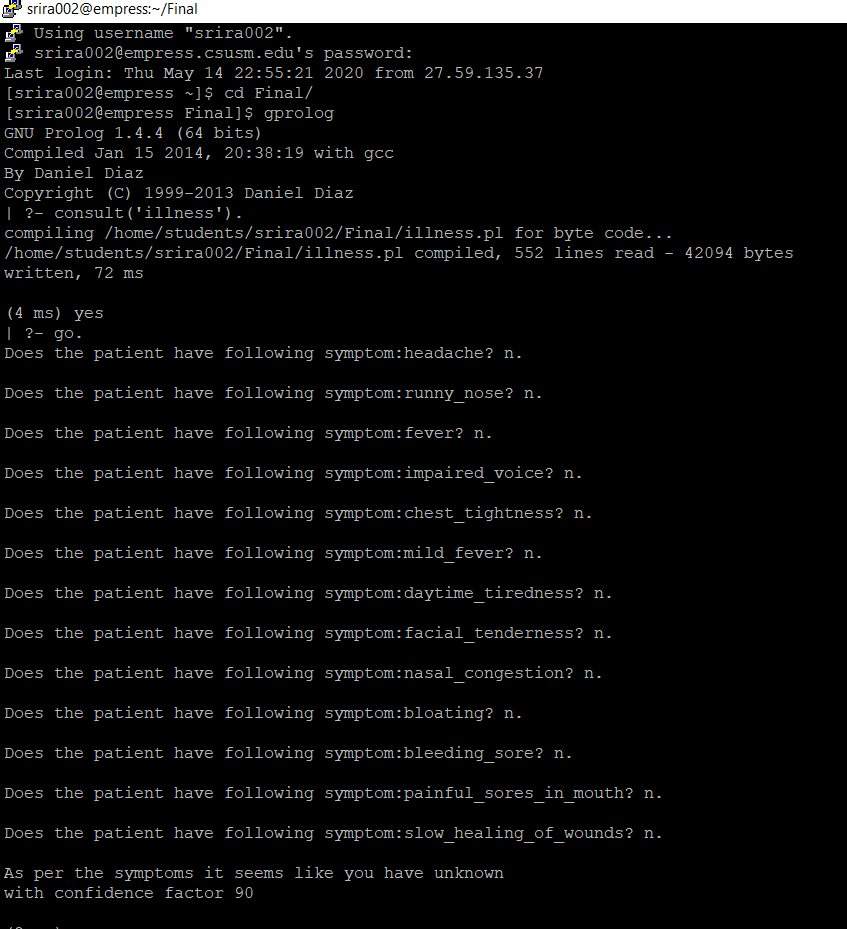
Undo Rule

|  |
| --- |
| undo :- retract(yes(\_)),fail.  undo :- retract(no(\_)),fail. |

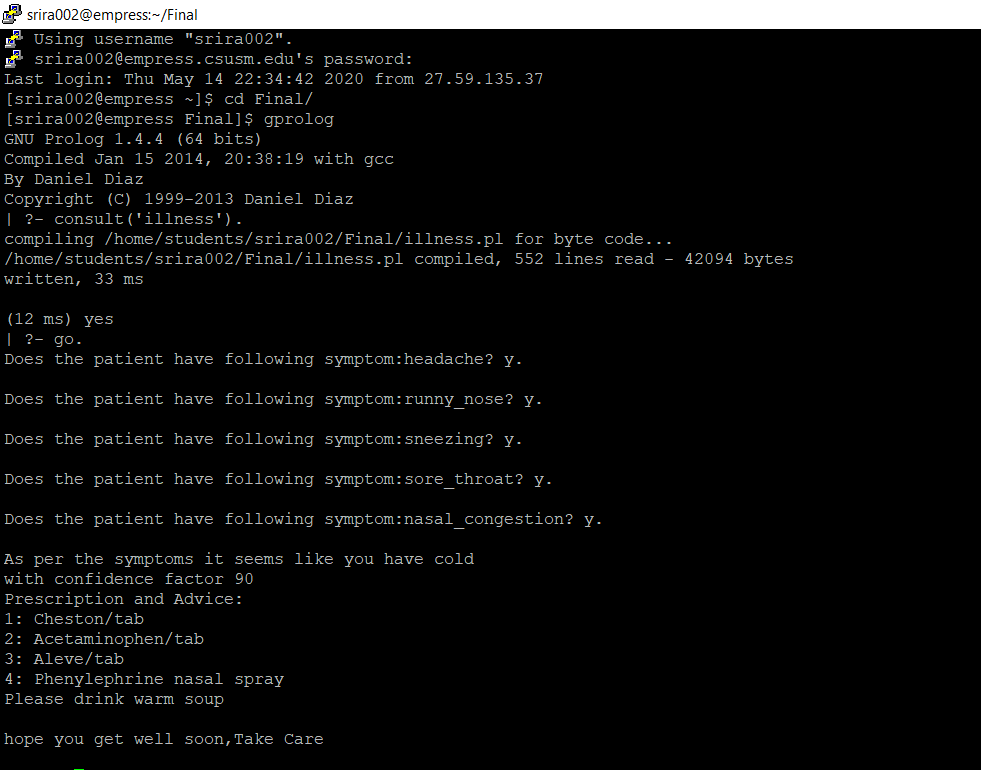
This rule is used to undo all yes no assertions i.e. to retract the user answers which is true only when the right side is true else false.

**Section 3: Testing and Results**

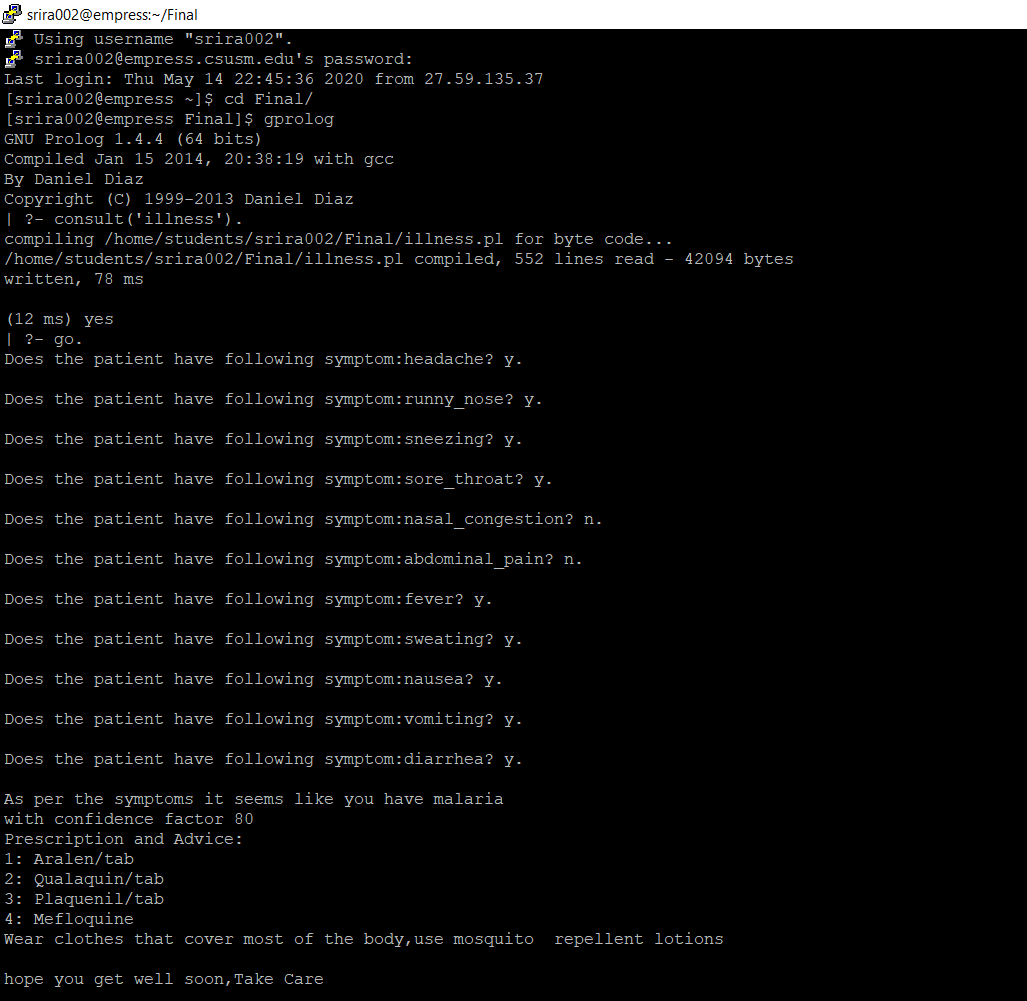
**Test 1:**



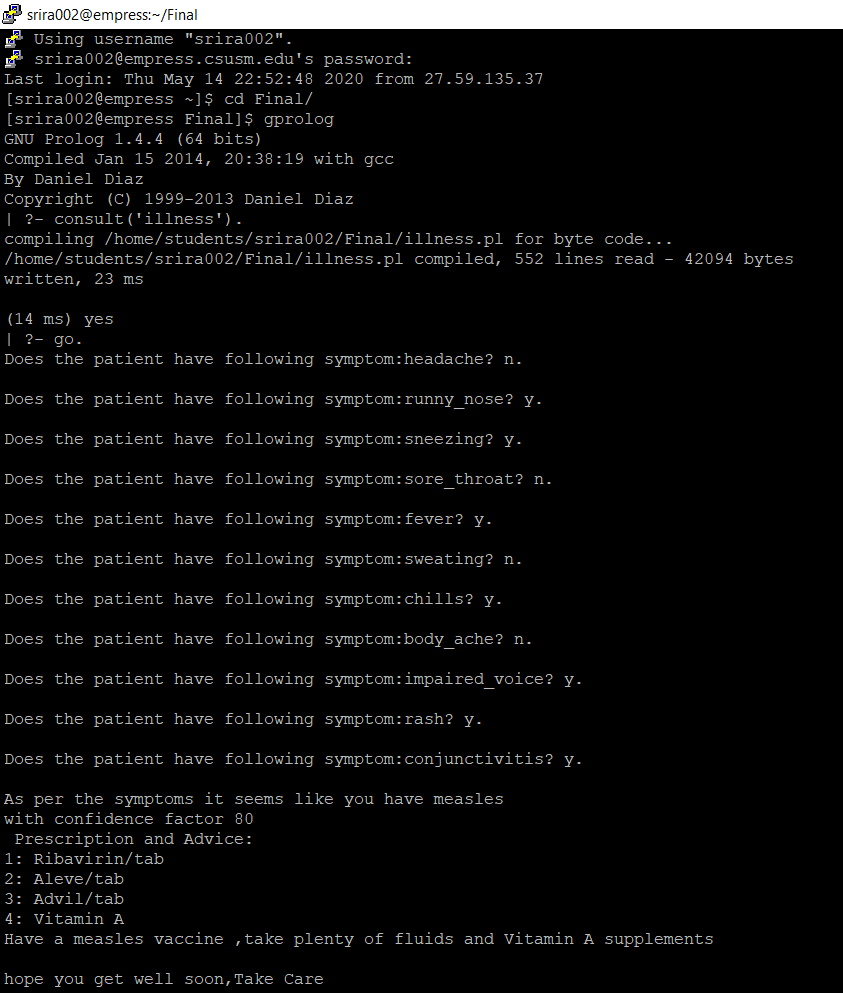
**Test 2:**



**Test 3:**

****

**Test 4:**

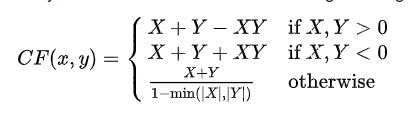


**Section 4: Confidence Analysis**

As per the MYCIN we know that we can have either two or more rules to conclude a parameter which has different weights of evidence.

For example in the Diagnose Illness System we can see that one rule would conclude that user has chicken pox with a confidence factor of 90 .i.e. 0.9and another rule would conclude that the user has chickenpox with a confidence factor of 80 i.e.0.8.

So to calculate certainty factor according to MYCIN we need to combine these weights by using the formula.



Here X and Y represent the confidence factor of the rules.We can apply this same formula to calculate the confidence if we have more than two rules which conclude the same parameter but with different confidence factors.The order in which the weights are combined would not matter as this is commutative.

As both X and Y are positive in our case we can say that the confidence factor is

(0.9+0.8-(0.9\*0.8))

= 0.98

Therefore, the confidence measure for the expert system is 0.98 if we take an example of cold.

In Test 3 we see that malaria is concluded So

assumption((malaria),80) :-

check(fever),

check(sweating),

check(headache),

check(nausea),

check(vomiting),

check(diarrhea).

// let us consider certainty factor for Rule is 80(0.8) and fever is 0.98, sweating is 0.90, headache is 0.96, nausea is 0.97, vomiting is 0.98 and diarrhea is 0.96

Certainty Factor = Min(0.98,0.90,0.96,0.97,0.98,0.96) \* (0.8)

= 0.90 \* 0.8

= 0.72

For Test4:

We conclude measles So

assumption((measles),80) :-

check(fever),

check(runny\_nose),

check(rash),

check(conjunctivitis).

//let us consider certainty factor for Rule is 80(0.8) and fever is 0.98, runny\_nose is 0.75, rash is 0.97, conjunctivitis is 0.98

Certainty Factor = Min(0.98,0.75,0.97,0.98) \* (0.8)

= 0.75\*0.8

= 0.6

Similarly, we can calculate and conclude others.

**Section 5: Machine Learning**

We can incorporate machine learning into rule based expert systems by using a machine learning algorithm which uses the datasets of the medical field to classify into something and this information from the machine learning algorithm can be used to form the new rules and facts i.e the knowledge base of the system and we can also use artificial neural nets to learn from the expert system by updating the weights i.e confidence of the rules and the facts of the rule based system to achieve better results and to develop better expert systems.Also We can incorporate machine learning into our system in some ways like we can use a machine learning algorithm to classify illness based on the system and then we can use this information from the machine learning to form rules and facts i.e. symptoms and rules for the knowledge base of the system. We can also use neural nets to update the certainty factors of the rules which we conclude to provide better results.

Also we can use the machine learning algorithm to work on a different dataset i.e. a better dataset from the previous dataset and this would allow us to make new/better rules and facts for the knowledge base of the system and we can also use the user input to make new rules and facts to provide better results using neural nets.

This would be very useful because there would be very little differences in the symptoms to classify between illnesses so such a system which would learn and use the learning outcomes from training by using different datasets and form new rules and facts i.e. knowledge base of the system, instead of sticking with the same rules and facts which need to be changed as the illnesses and symptoms changes frequently be changing. By learning from these changes we can always build a better illness diagnosis system.Also we can add new symptoms and illnesses to the knowledge base of the system using the learning so we need to provide new prescriptions and advice to the user.One such example would be when the user does not have any of the symptoms asked by the expert system then it can use new dataset which has more data about the symptoms and illness then the algorithm can be used to form new rules and facts from symptoms and diseases and this would now ask some new questions about the system to the user and this could be what the user is experiencing.

**References**

[**https://en.wikipedia.org/wiki/Mycin**](https://en.wikipedia.org/wiki/Mycin)

[**https://www.researchgate.net/publication/221438964\_Hybrid\_Approach\_Combining\_Machine\_Learning\_and\_a\_Rule-Based\_Expert\_System\_for\_Text\_Categorization**](https://www.researchgate.net/publication/221438964_Hybrid_Approach_Combining_Machine_Learning_and_a_Rule-Based_Expert_System_for_Text_Categorization)

[**https://www.researchgate.net/publication/271157162\_Medical\_Expert\_Systems\_for\_Diagnosis\_of\_Various\_Diseases**](https://www.researchgate.net/publication/271157162_Medical_Expert_Systems_for_Diagnosis_of_Various_Diseases)

http://www.ijmlc.org/vol5/492-A14.pdf

<https://www.cpp.edu/~jrfisher/www/prolog_tutorial/2_17.html>