

**CZ3005 Artificial Intelligence**

**Assignment Report**

**Submitted by Yong Hao**

**Matriculation Number: U1722282A**

**Lab Group: TSR1**

School of Computer Science & Engineering

**1. Abstract**

The report describes a program written in SWI-prolog declarative language to mimic the diagnosis process by conversing with a patient who can only say yes or no, and additional two commands to quit the program and force to diagnose. The above functionality is required by the fourth assignment in the “Artificial Intelligence Assignments” file, and the title of the assignment is “Patient with a sympathetic doctor”.

The doctor will ask about the mood and pain level of the patient, and the tone of the questions will change accordingly. Afterwards the doctor will ask consecutive questions to determine the symptoms the patient has. If the patient affirms one particular symptom, the symptom will be recorded, and another related symptom will be randomly picked, and the program will ask the patient again. Otherwise if the patient negates the symptom, the program will record it as well and randomly pick one symptom from all available unasked symptom poll to ask the patient.

If one or more diseased can be diagnosed, the program will display the diagnose analysis and terminate. If no match can be found, the program will print the error message and terminate. The diseases and their related symptoms are pre-defined and stored in the program’s knowledge base. The program can be rerun for arbitrary times in one execution process with previous diagnosis history cleared.

**2. Algorithm Explanation**

**2.1. Predefined Knowledge Base**

**2.1.1. Mood and Pain List**

****

5 types of mood and 5 levels of pain are chosen and each of the mood and pain are associated with a seriousness degree. The higher the degree, the more serious and mentally and physically unstable the patient is. The degrees will function as a mathematical model in later predicates to determine the tone of the doctor. In our program, we take x = 0.2\*mood + 0.8\*pain to be the overall seriousness of the patient.

**2.1.2. Doctor’s Gesture List**

****

The above facts model the possible tones the doctor might express in the diagnosis process. Depending on the seriousness degree, the doctor will choose one gesture in either one of the three gesture lists. For example, if the overall serious is higher than 3, the doctor might initiate a question as below:



**2.1.2. Diseases and Symptoms List:**

****

5 diseases which reside relatively far away from each other in the human disease spectrum are modelled with their symptoms. Two symptoms under one disease are considered related. The following predicate is written to determine the “relationship” between two symptoms:



**2.1.3. Dynamic Knowledge Base and Initialization**



askedsymp([]) initially contains one empty list. Along with the diagnosis process, all asked symptoms will be updated here to prevent from being asked again.

Similar to askedsymp([]), has([]) stores all symptoms the patient confirms and hasnot([]) stores all symptoms the patient negates. The list in has() will be utilized to diagnose the patient. The list in hasnot() has no practical use in the program, but can be easily extended for more advanced expansion.

The default mood and pain level for patient are “calm” and “no pain”. These will be updated when the diagnosis process begins.

**2.2. Diagnosis Workflow**

**2.2.1. Begin and Initialize**

****

Enter the goal **begin().** to start the program. This predicate clears the information of the previous patient and initialize has(), hasnot(), and askedsymp(). A welcome message will be printed. Then it gets the moodList L and enter the predicate askMood(L) with L.

**2.2.1. Get Mood from Patient**

****

SympatheticAsk() is a procedure to determine the tone based on the overall seriousness of the patient. See section 2.3. for details.

List\_empty(L, V) is a procedure to determine if the list is empty. If L is empty, V will be true, and vice versa. See section 2.4. for details.

The procedure will firstly choose a tone for the doctor, and if L is empty, terminate the program since all mood choices have run out. Otherwise, the head of L (call it H) will be asked. If patient has mood H, diagnoseMood(H) will update patientmood(), and get painlist(L) and enter procedure askPain(L). If not, the procedure recursively calls itself with the rest of the mood list.

If user enters ‘q’, the program will be forced to terminate. And if user enters ‘d’, the program will be forced to diagnose.

**2.2.2. Get Pain from Patient**



The mechanism is similar to askMood. askPain(L) will update patientpain according to user input. And will ask the user if he/she has “cough” to begin the askSymptom(S) procedure.

**2.2.3. Ask Symptoms of Patient**

****

SympatheticAsk() will be called first. Depending on the user input, either affirmSymptom or negateSymptopm will be called.

**2.2.4. Affirmation and Negation of Symptom**

****

If the patient affirms a symptom, it will be updated in has() and askedsymp() respectively, then a list that contains all symptoms related to X into a list Y. A random symptom in Y will be asked, if Y is not empty. Otherwise it means all symptoms of one disease have been depleted and the diagnosis will begin.

If the patient affirms a symptom, it will be updated in has() and askedsymp() respectively, then a list that contains all symptoms from all diseases will be generated. A random symptom from the all symptom pool and is not a member of asked symptoms (R) will be picked and asked. If R is empty, it means all symptoms have been asked and the diagnose will begin.

**2.2.5. Diagnose Session**

****

diagnoseBegin() will print a message and enter diagnose procedure.

diagnose() will determine the disease the patient has by calling individual diagnose procedure and enter diagnoseEnd().

diagnoseEnd() will print a message and abort the session.

**2.2.6. Disease Diagnose**

Take diagnosecold(). as an example:



Procedures of this kind will get the list in has() and intersect with all the symptoms a disease has. If more than 4 symptoms are matched, the diagnosis will be printed and the diagnose session would end.

**2.3. Sympathetic Gesture Choosing.**

****

Procedure sympatheticAsk() will get the seriousness degree of the patient and pass the overall seriousness (D\*0.2+S\*0.8) to chooseTone(D). Depending on D, one of the gesture procedure will be called and one gesture will be chosen for the doctor.

**2.4. Utilization Predicates**

****

To check if a given list is empty.

**3. Appendix**

The following files are enclosed in the submission zip file:

1. CZ3005\_Assignment\_Report\_Sympathetic\_Doctor.docx – This report.

2. sympathetic\_doctor.pl – The SWI-Prolog program of the sympathetic doctor.

3.demo.mp4 – which is the demo video containing the following test cases:

1> Patient has a calm mood and feels not pain – test doctor’s gesture;

2> Patient has a melancholic mood and feels intolerable pain – test doctor’s gesture;

3> Diagnose with cold by affirming 5 symptoms.

4> Diagnose with cold by affirming 4 symptoms and force diagnose.

5> Quit program in between.

6> Affirm three symptoms of cold and force diagnose.

7> Diagnose with diabetes

8> Negate all symptoms.

Please refer to the video for detailed demonstration