

**CZ3005: Artificial Intelligence**

**2020 Fall Assignment 4:**

**Implementing a Talking Box with Prolog**

**Question 4:**

**Patient with a Sympathetic Doctor**

**Name: Royce Ang Jia Jie**

**Matriculation Number: U1840416D**

**Tutorial Group Number: TSP2**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**2020/2021 SEMESTER1**

Table of Contents

[Section 1. Introduction 3](#_Toc56728017)

[Section 2. Overview 3](#_Toc56728018)

[2.1 Introduction to KBS 3](#_Toc56728019)

[2.2 Overview of KBS 4](#_Toc56728020)

[2.3 Logic flow of Sympathetic Doctor Program 5](#_Toc56728021)

[5](#_Toc56728022)

[2.4 Further Considerations 6](#_Toc56728023)

[**2.4.1 Pain Level and its effect on the biasness of diagnosis** 6](#_Toc56728024)

[**2.4.2 Pain and Mood Level & their effect on the degree of sympathy** 6](#_Toc56728025)

[Section 3. Implementation and Explanation 7](#_Toc56728026)

[3.1 Code Explanation 7](#_Toc56728027)

[3.2 Logic Flow of Main Program 7](#_Toc56728028)

[3.3 Logic Flow for Diagnosis 9](#_Toc56728029)

[3.4 Heuristics/ Algorithm for conflicting diagnosis 11](#_Toc56728030)

[3.5 Knowledge Base 12](#_Toc56728031)

[3.6 Helper Functions 12](#_Toc56728032)

[3.7 Further Considerations (Error Handling) 12](#_Toc56728033)

[Section 4. Demonstration of Sympathetic Doctor Dialogue AI 13](#_Toc56728034)

[Section 5: Additional Implementation (GUI) with Google Speech Recognition Engine 13](#_Toc56728035)

[5.1 Motivation for GUI Implementation (Additional Feature) 13](#_Toc56728036)

[5.2 Overall System Architecture Diagram of the GUI Implementation 13](#_Toc56728037)

[5.3 Set up and Installation 14](#_Toc56728038)

[5.4 Implementation of the GUI 14](#_Toc56728039)

[5.5 Screenshot of GUI Implementation 14](#_Toc56728040)

[5.6 Demonstration of Sympathetic Doctor Dialogue AI with GUI (YouTube Link) 14](#_Toc56728041)

[Section 6. Conclusion 15](#_Toc56728042)

[Works Cited 16](#_Toc56728043)

# **Section 1. Introduction**

Prolog is a logic programming language associated with Artificial Intelligence and Computational Linguistics.

It is interesting to note that unlike other programming languages, Prolog has its roots in first-order logic, a formal logic, used to represent a powerful Knowledge-Based System (KBS). As such, logic programmers can ascribe the efficiency of the knowledge-based representation to its generic representation without having to deal with technical concepts (time, events). Additionally, it is a universal language, allowing us to express anything that can be programmed.

In this assignment, we will:

(1) Implement a *Knowledge Base System Dialogue AI* to explore the concept of separation of *Knowledge* (essential representation of the world) from the *Inference Engine* (Set of inference rules) in KBS using Prolog.

(2) As part of an additional feature, we will be implementing a *GUI* to simplify and have an active interaction with the KBS, while observing how the KBS does the update in the background.

# **Section 2. Overview**

## **2.1 Introduction to KBS**

The KBS is designed with the following considerations:

1. The KBS must be able to ask user(patient) questions.
2. The user(patient) must be able to reply “yes” or “no” only.
3. The KBS must be able to ask the user(patient) 5 or more degree of pain level.
4. The KBS must be able to ask the user(patient) 5 or more levels of mood.
5. The KBS must be able to ask questions in an appropriate demeanour based on the user’s(patient) pain and mood level.
6. The KBS must have a total of 5 diseases.
7. For each disease, there must be at least 5 or more symptoms associated to it.
8. The KBS must be able to diagnose the patient’s disease.

Some additional considerations during the design of the KBS:

1. When a user(patient) replies a “yes” to any question, KBS must be able to *assert* the given answer as true.
2. The KBS is assumed to be a *thorough* AI. In which, the KBS should be able to do a *comprehensive analysis* of various symptoms should be done before making a diagnosis.
   1. KBS should be able to iterate through all symptoms.
   2. A good heuristic or algorithm should be implemented for diseases with overlapping symptoms.
3. The KBS must be able to answer the user(patient) in a sympathetic manner.
   1. A combination of pain level and mood level should output a type of gesture.
   2. Gesture chosen should be able to distinctively show an appropriate level of sympathy.

## **2.2 Overview of KBS**

To be able to separate the knowledge base from the inference engine, the KBS has the following 5 distinct “Databases”. To ensure that the symptoms of each disease/illness is as *realistic* as possible, I have referenced them from NHS disease database [1].

A picture containing graphical user interface

Description automatically generated

Figure 1 5 Distinct "Databases"

## **2.3 Logic flow of Sympathetic Doctor Program**

## A picture containing shape Description automatically generated

Figure 2 Sympathetic Doctor Logic Flow

Firstly, the Doctor will ask the patient if he/she is feeling any pain?

e.g. Doctor : “Hello! I am Doctor Box. Do you feel any pain?”

If the patient replies a “*No*”, his reply would implicitly assert that the patient is not having pain or “***pain free***”.

Otherwise, if the patient replies a “*Yes*”, this would initiate the iteration of the remaining list of 4 pain questions, until it is empty.

e.g. Doctor: “*Do you feel* ***mild pain***” (*no*)🡪 “ *Do you feel* ***moderate pain***” (*no*)🡪

Doctor: “*Do you feel* ***severe pain***” (*no*)🡪 “*Do you feel* ***overwhelmingly severe pain***”   
 (*no*)🡪 {*Empty List*}

Edge cases such as having an empty list is handled by implicitly asserting that the patient is not having pain or “***pain free***”.

The same procedure is repeated for asking *Patient’s Mood Level* (*Calm, Worried, Stressed, Fearful and Panic Stricken*).

The combination of pain level and mood level would determine how the Doctor should respond to the patient from one of the 9 varying types of gestures ( *Humorous, Attentive, Accommodating, Amiable, Very Attentive, Console, Comfort, Reassure and Companion*).

e.g. Overwhelming Pain + Panic Stricken 🡪 Companion Gesture

Doctor: “***Let us go through this together*** (*Companion*). Do you have a cough? (*Symptom*)”

## **2.4 Further Considerations**

### **2.4.1 Pain Level and its effect on the biasness of diagnosis**

To simulate a real-life situation, where a doctor would use pain level to bias towards a specific type of diagnosis.

An additional feature is added such that pain level would also induce the biasness of diagnosis. This is implemented by **adding** *additional weights* (*heuristic*) to *diseases* associated to specific pain level; thereby, giving these disease(s) a heard start.

Note: This will be explained in the *implementation and explanation section*.

A picture containing shape

Description automatically generated

Figure 3 Weighted Disease(s) Associated to specific Pain Level

### **2.4.2 Pain and Mood Level & their effect on the degree of sympathy**

Another challenge faced was the designing of gestures. A great emphasis is placed on good patient-physician communication skill to ensure the authenticity of the dialogue with the Doctor [2]. This is done by combining both non-verbal (e.g. *emoticons*) and verbal cues (e.g*. filler words*) for a single gesture.

e.g. *gesture(humorous) :- knowledgeable(A), emoticons(B), kidding(C), filler words(D)*

Note the mood level increases in the following order:

***Calm*** *<* ***Worried*** *<* ***Stressed*** *<* ***Fearful*** *<* ***Panic Stricken***

The degree of seriousness in gestures are as follow:

***Humorous*** *<* ***Amiable*** *<* ***Attentive*** *<* ***Very Attentive*** *<* ***Accommodating*** *<* ***Console*** *<* ***Comfort*** *<* ***Reassure*** *<* ***Companion***

Additionally, because of the extreme case, where a patient may experience (1) ***Overwhelmingly Severe Pain*** and (2) ***Calm Mood***, it would be inappropriate to have a gesture of ***Humorous***, but a more (in-between) serious degree of gesture should be picked instead, for instance, the gesture of ***Comfort****.*

Also, (1) ***Overwhelmingly Severe Pain*** *and (2)* ***Panic Stricken*** *🡪* ***Companion*** *Gesture*

# **Section 3. Implementation and Explanation**

## **3.1 Code Explanation**

## **3.2 Logic Flow of Main Program**

Step 1: Initialization of Main Program via ask(0).

Text

Description automatically generated

Step 2: Selects pain question from pain question ‘*Database*’ and return back to ask\_repeat program.

Text

Description automatically generated

Step 3: When pain level has been selected, proceed to query the level of mood.

Text

Description automatically generated

Step 4: Selects mood question from mood question ‘*Database*’ and return back to ask\_mood\_repeat program.

Text

Description automatically generated

Step 5: When mood level has been selected, proceed to query\_symptoms program.

Text

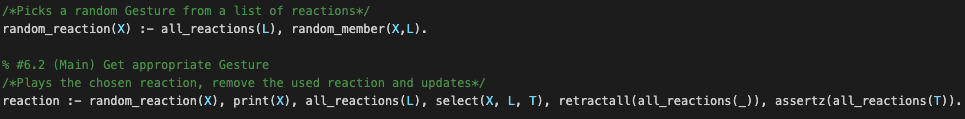
Description automatically generated

Step 6.1: Query for a random symptom and with the appropriate gesture from the Doctor, ask the patient sympathetically.

Text

Description automatically generated

Step 6.2: Selects an appropriate Gesture before returning to *Step 6.1*  to ask the patient sympathetically.

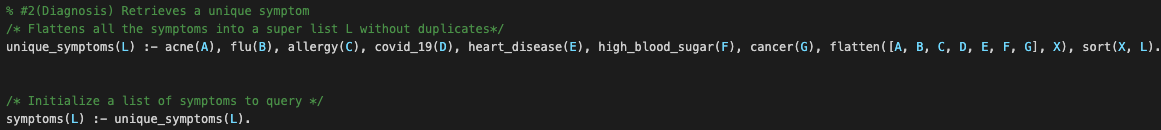


## **3.3 Logic Flow for Diagnosis**

Step 1: Query for random symptom



Step 2: Selects a unique symptom for Step 1



Step 3: If patient answers “*Yes*” to a symptom, assert that symptom and proceed to merge(symptom) program.

Text

Description automatically generated

Step 4: Assert the appropriate symptom and increment the symptom accordingly.

Text

Description automatically generated

Step 5: (*This is only a snippet, please refer to the code for the full implementation*) Increments Disease Counter

Text

Description automatically generated

Step 6: (*This is only a snippet, please refer to the code for the full implementation*)

Text

Description automatically generated

Step 7: When symptom list is empty, proceed to diagnose program. Disease with the *highest heuristic* will be considered for the diagnosis of the patient.

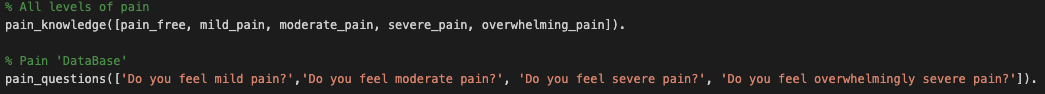
Text

Description automatically generated

## **3.4 Heuristics/ Algorithm for conflicting diagnosis**

## **3.5 Knowledge Base**

Pain Database (Note: ‘pain\_free’ question is asked in ask(0) program of “Do you feel any pain?”)



Mood Database



## **3.6 Helper Functions**

Text

Description automatically generated

## **3.7 Further Considerations (Error Handling)**

Rules governing pain query empty list (error handling).

**Text

Description automatically generated**

Rules governing mood query empty list (error handling).

Text

Description automatically generated

# **Section 4. Demonstration of Sympathetic Doctor Dialogue AI**

# **Section 5: Additional Implementation (GUI) with Google Speech Recognition Engine**

## **5.1 Motivation for GUI Implementation (Additional Feature)**

## **5.2 Overall System Architecture Diagram of the GUI Implementation**

A picture containing shape

Description automatically generated

Figure 4 System Architecture Diagram of Implemented GUI

## **5.3 Set up and Installation**

## **5.4 Implementation of the GUI**

## **5.5 Screenshot of GUI Implementation**

## **5.6 Demonstration of Sympathetic Doctor Dialogue AI with GUI (YouTube Link)**

# **Section 6. Conclusion**

# Works Cited

|  |  |
| --- | --- |
| [1] | A. K. A. C. Piyush Ranjan, "How can Doctors Improve thier Communication Skills?," *NCBI,* vol. 9, no. 3, pp. 01-04, 2015. |
| [2] | NHS, "NHS inform," [Online]. Available: https://www.nhsinform.scot/illnesses-and-conditions/immune-system/allergies. [Accessed 19 November 2020]. |