Harry James Hall – C3550862

Leeds beckett university  Headingley Campus

mEDICAL DIAGNOSIS SYSTEM FOR DISEASES OF THE IMMUNE SYSTEM

# Abstract:

Immune diseases can be divided into into two broad categories: Autoimmune where the immune system attacks the body; and immune dysfunction where the immune system cannot properly defend the body against pathogens.

Auto-immune diseases include things like Lupus, Allergies, and Paraneoplastic syndromes.

Immune dysfunction includes diseases like HIV and AIDS caused by environmental factors and diseases like Chronic Granulomatosis Disease which are caused by genetics.

This system aims to help doctors or medical professionals diagnose both types of disorder based on a patients symptoms or test results.

Ideally this system would be designed to be easy to use by doctors with limited computer literacy, but this might not be achievable using only Prolog as the language the system is written in as it appears to have many limitations for making interfaces, even something like a CLI.

## Aims and objectives:

1. Provide a system that is useful in medical diagnosis to doctors looking for immune related diseases such as specialists
2. Make the system easy enough to use for doctors with minimal training

# Literature review and similar systems:

There is a medical expert system based on IBM’s Watson. IBM’s Watson is a AI system that was famous for answering questions on the American TV show Jeopardy.

It has the ability to understand questions given to it in English and answer them using information from the internet and it’s knowledge base. It also has the ability to score answers that it finds based on available evidence to determine how likely the answer is to be correct.

<https://www.nist.gov/system/files/documents/healthcare/NIST-Siegel_PART-3.pdf>

A lot of modern medical diagnostic systems use something called fuzzy logic along with neural networks.

<https://ieeexplore.ieee.org/document/9279211>

# Design:

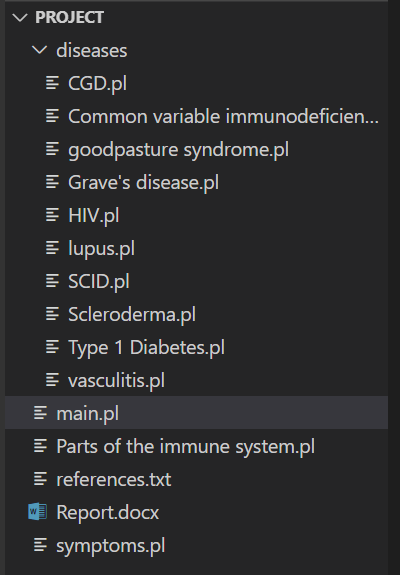
The system will first ask the doctor for a list of symptoms.

If the system can find diseases that match all of the symptoms it will display them, otherwise it will display diseases that match only some of the symptoms presented by the patient.

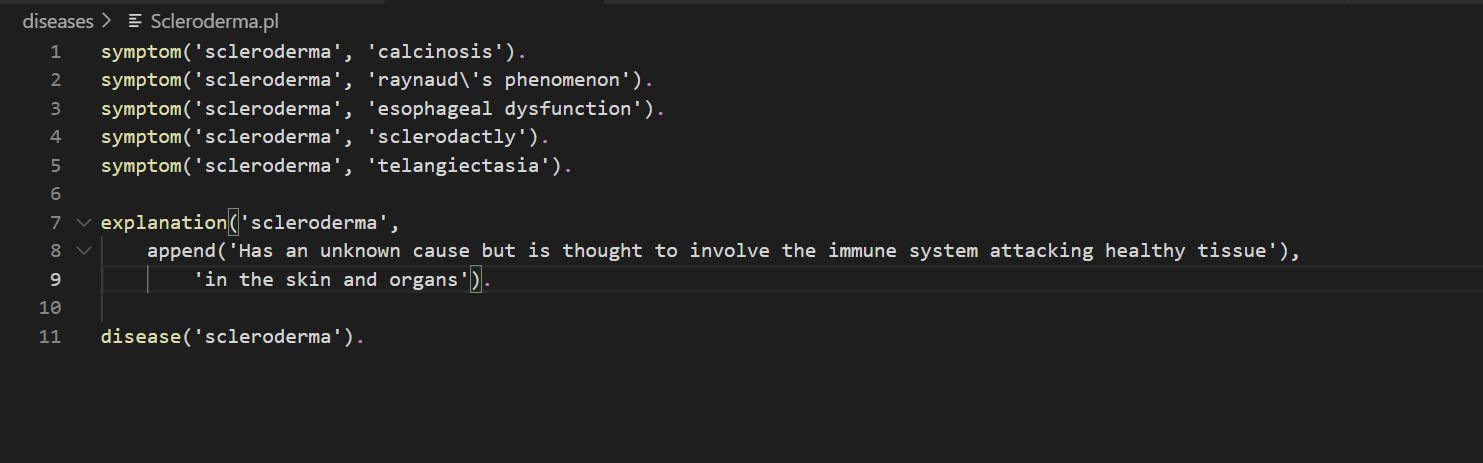
# Implementation:

The system stores it’s knowledge in separate files to the main logic.

The knowledge for each disease is stored in the diseases sub directory.

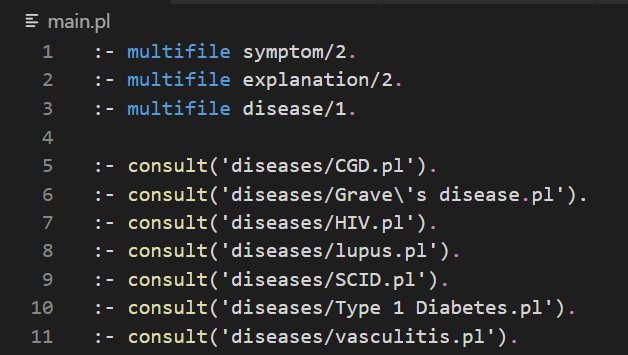


Each file has a list of symptoms stored in the symptom predicate:



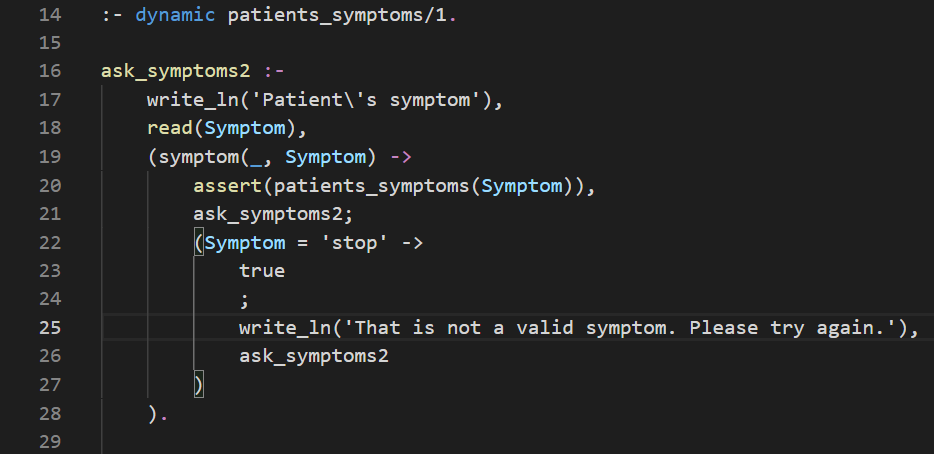
This also includes an explanation of the disease that can be printed out and the diseases name.

All of the disease files are consulted by the system at the beginning of main.pl.



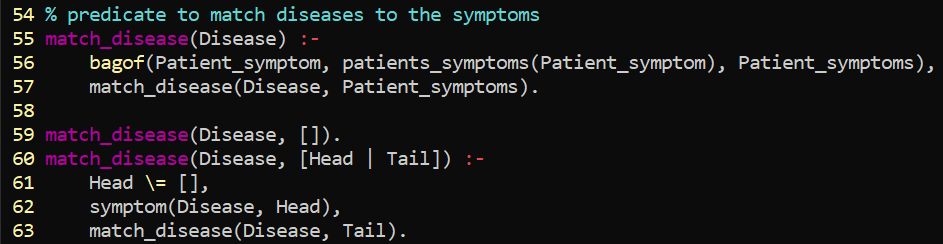
The predicates symptom, explanation, and disease have to be declared as multifile predicates because they are split across multiple files otherwise it gives an error.

The program asks the doctor about the symptoms the patient has using this predicate:



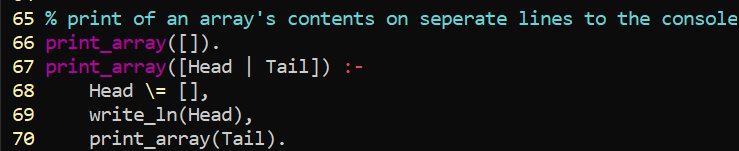
It stores the symptoms in a predicate called patients\_symptoms/1 using assert. It uses recusion to ask for more than one symptom and stops when someone types ‘stop’. It does this using an if statement with ->.

The program then begins finding diseases that match all of the symptoms using the match\_disease predicates.

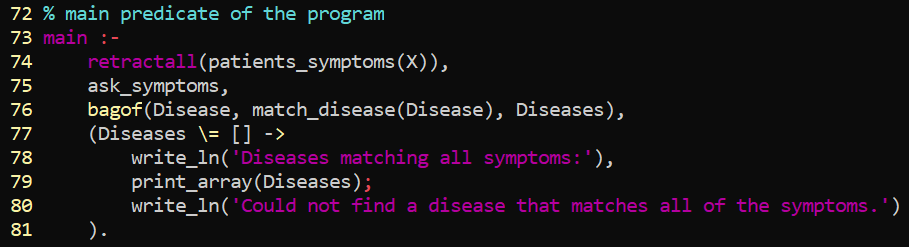


This predicate first gets a list of all the symptoms the doctor entered using the bagof/3 predicate. It then calls itself with that array as an argument. The third clause of match\_disease deals with this by matching the head of the list to the disease then calling itself again. The predicate returns true when the list is empty, this is the termination condition for the recursion. If it fails to match any symptoms it will fail immediately.

I defined a predicate to print out a list of items:



This also works using recursion printing out one item at a time using write\_ln/1 then calling itself on the tail of the list.



This first removes any existing symptoms from running the program more than once using retractall/1. It then calls the ask\_symptoms/0 predicate. It then gets a list of all matching disease using bagof/3 and match\_disease/1.

An if statement is then used to check if any diseases were found. It prints them out with print\_array/1 predicate that I defined