

Trabalho de Linguagens de Programação - Relatório

Dener Luis Basilio Theodoro - 201835001
Lásaro de Almeida Deodoro - 201835004

Dezembro 2023

1 Introdução

Neste trabalho para a disciplina de Linguagem de Programação, ministrada pelo professor Leonardo Reis, foi desenvolvido um bot clínico geral na linguagem Prolog. O objetivo deste trabalho foi aplicar os conceitos da linguagem Prolog apresentados em sala de aula.

2 *Clínico geral bot*

O objetivo deste trabalho é implementar, em Prolog, um *clínico geral bot* que dê diagnósticos rápidos e previos com base nos sintomas assinalados pelos pacientes. Para tal, o bot deverá realizar uma serie de perguntas acerca dos sintomas do paciente e inferir qual a doença do paciente e quais motivos levaram a esse diagnóstico.

Foi solicitado que o *clínico geral bot* fosse capaz de inferir sobre ao menos 20 doenças, com a restrição de que ao menos 15 pares das doenças escolhidas tivessem ao menos 3 sintomas em comum. Ainda, o programa não deve realizar a pergunta de um sintoma mais de uma vez e, ao final, a razão para o diagnóstico deve ser exibida.

Um exemplo da execução é dado a seguir:

?- especialista(X).

apresenta febre?

—: yes.

apresenta calafrios?

—: yes.

apresenta tosse?

—: yes.

apresenta dor de garganta?

—: yes.

apresenta dor de cabeça?

—: yes.

apresenta congestão nasal?

—: yes.

apresenta dores musculares?

—: no.

apresenta problemas no olfato ou no paladar?

—: yes.

Seu diagnóstico é: covid

A doença foi diagnosticada dado que o paciente relatou possuir os seguintes sintomas: febre; calafrios; tosse; dor de garganta; dor de cabeça; congestão nasal; problemas no olfato ou no paladar

X = covid .

3 Desenvolvimento

Nesta seção será apresentada como o *clínico geral bot* foi desenvolvido.

3.1 Estrutura utilizada

O código foi desenvolvido com base em um exercício dado em sala de aula, com alguns acréscimos, exercício o qual perguntas são feitas para se inferir um dos 3 animais possíveis: cachorro, gato e pato. Era possível inferir o animal com base em perguntas sobre as características e sons que fazem.

O código pode ser separado em 6 partes:

- Definição da precedência das operações;
- Definição das doenças e seus sintomas;
- Tratamento das entradas e retorno da doença inferida;
- Questionamento acerca dos sintomas;
- Resposta detalhada acerca do porquê a doença apresentada foi diagnosticada;
- Inicialização do *clínico geral bot*;

3.1.1 Definição da precedência das operações

A definição da precedência das operações é importante para o correto funcionamento das operações que desejamos.

```
:- op(800, yfx, if).  
:- op(200, xfy, and).
```

Neste trecho, é definido para 'if' uma precedência de 800 e uma associatividade à esquerda. Isso indica que os trechos à esquerda de 'if' tem menor precedência, enquanto os à sua direita tem maior precedência. Neste trecho é também definido para 'and' uma precedência de 200 e uma associatividade à direita. Isso indica que os trechos à esquerda de 'nd' tem maior precedência, enquanto os à sua direita tem menor precedência.

3.1.2 Definição das doenças e seus sintomas

O 'clause' foi usado para definir regras associadas à doenças e, usando o fator das precedências já definidas, foi definido o nome da cláusula (dado pelo nome da doença) e seus sintomas.

```
clause(doenca(covid) if  
'febre' and  
'calafrios' and  
'tosse' and  
'dor de garganta' and  
'dor de cabeça' and  
'congestão nasal' and  
'problemas no olfato ou no paladar').
```

No trecho acima é dado um exemplo da definição de uma doença. Nele em específico é definida uma cláusula para a doença covid e assinalado os sintomas correspondentes à doença. Isso é feito para outras 24 doenças. Ao fazer isso nos preocupamos em deixar as doenças com sintomas iguais descritos da mesma forma. Um exemplo pode ser dado pelo sintoma 'mau estar', que aparecia hora com espaço, hora com hífen, para separar as duas palavras. Optamos por deixar todas suas ocorrências separadas por espaço.

3.1.3 Tratamento das entradas e retorno da doença inferida

O trecho a seguir é o responsável por tratar as entradas e retorno a doença inferida:

```
solve(true).  
solve(X and Y) :- solve(X), solve(Y).  
solve(X) :- clause(X if Y), solve(Y).
```

Esse trecho retorna verdadeiro se a conjunção das cláusulas referente aos sintomas for verdadeira e, se verdadeira, ela retorna a doença cuja todas as cláusulas são verdadeiras.

3.1.4 Questionamento acerca dos sintomas

Foi elaborada uma pergunta para cada sintoma presente nas doenças cadastradas, tomando cuidado para que todos os sintomas fossem questionados e que cada um só fosse perguntado ao paciente uma única vez. Um exemplo de pergunta pode ser vista a seguir:

```
question :- write('apresenta falta de apetite?'), read(yes), asserta('falta de apetite' if true)).
```

As perguntas são feitas até que se ache uma doença compatível com os sintomas indicados pelo paciente. O 'write' é responsável por apresentar a pergunta na tela, o 'read' é responsável por ler a resposta do paciente e, se o paciente digitar 'yes', o sintoma perguntado é declarado verdadeiro (declarado como um predicado verdadeiro, papel de 'clause') e, caso todas os sintomas de uma doença forem demarcados como 'yes', as perguntas param de ser exibidas e o diagnóstico é concluído.

3.1.5 Resposta detalhada acerca do porquê a doença apresentada foi diagnosticada

'imprimir_sintomas' é responsável por, dado um diagnóstico, imprimir a doença diagnosticada junto aos sintomas que levaram ao diagnóstico desta doença. Para imprimir corretamente os sintomas das doenças foi criado o 'doencaX' para cada doença, o qual, dada uma doença, imprime os sintomas dessa doença.

O trecho 'imprimir_sintomas' e um exemplo de 'doencaX' para a doença anemia é dado a seguir:

```
imprimir_sintomas(Diagnostico) :-  
format('Seu diagnóstico é: w n', Diagnostico),  
doencaX(Diagnostico).  
  
doencaX(anemia) :-  
write('A doença foi diagnosticada dado que o paciente relatou possuir os seguintes sintomas:  
fraqueza; indisposição; falta de ar; tontura; palpitação; dor de cabeça; palidez de pele e mucosas;  
enfraquecimento dos cabelos e unhas').
```

3.1.6 Inicialização do *clínico geral bot*

O comando 'specialista(X)' é responsável por iniciar o *clínico geral bot*. Dado o comando, uma série de perguntas são feitas, onde o paciente deve responder 'yes' para sintomas que possui ou qualquer outra coisa para sintomas que não possui. Essa parte é feita pelo 'question'.

O 'solve' é responsável por testar se as combinações dos sintomas se enquadram em alguma doença e, se sim, esta doença é salva em uma variável X. Por fim, esta variável é passada para a função 'imprimir_sintomas' para que seja dado o diagnóstico final. O trecho deste código é apresentado a seguir:

```
especialista(X) :- question, solve(doenca(X)), imprimir_sintomas(X).
```

4 Como executar o *clínico geral bot*

Para executar o *clínico geral bot* é preciso abrir no terminal a pasta que contém o arquivo "clinico_geral_bot.pl". Execute o arquivo com o comando:

```
swipl clinico_geral_bot.pl
```

Depois, para começar o *clínico geral bot*, chame a função especialista(X):

```
especialista(X).
```

A partir disso é só responder 'yes' para os sintomas apresentados pelo paciente, ou qualquer outra coisa para os sintomas que o paciente não apresenta.