



Sistemas de Representação de Conhecimento e Raciocínio

Avaliação em Grupo - 2ª Fase

Grupo 20

Guilherme Martins a89532 - Jaime Oliveira a89598 - João
Pereira a89607 - José Costa a89519 - Tiago Freitas a89570



a89532

a89598

a89607

a89519

a89570

Mestrado Integrado em Engenharia Informática

3º Ano - 2º Semestre

Resumo

Este documento refere-se ao segundo exercício da Unidade Curricular de Sistemas de Representação de Conhecimento e Raciocínio.

A realização deste exercício pretende motivar-nos para a utilização da extensão à programação em lógica, usando a linguagem de programação em lógica PROLOG, no âmbito da representação de conhecimento imperfeito, recorrendo à utilização de valores nulos e da criação de mecanismos de raciocínio adequados.

Índice

1	Introdução	4
2	Preliminares	5
3	Descrição do Trabalho e Análise de Resultados	5
3.1	Conhecimento negativo	5
3.2	Conhecimento imperfeito	6
3.2.1	Tipo incerto	6
3.2.2	Tipo impreciso	8
3.2.3	Tipo interdito	9
3.3	Evolução do conhecimento	11
3.3.1	Conhecimento incerto	11
3.3.2	Conhecimento impreciso	12
3.3.3	Conhecimento interdito	13
4	Conclusão	14

Figuras

1	Exemplo de conhecimento negativo	5
2	Predicado nao	6
3	Predicado demo	6
4	Exemplo de utilização do demo	6
5	Conhecimento incerto: utente	7
6	Conhecimento incerto: centro de saúde	7
7	Conhecimento incerto: staff	7
8	Conhecimento incerto: vacinação	7
9	Exemplo de conhecimento incerto do staff	8
10	Conhecimento impreciso: utente	8
11	Conhecimento impreciso: centro de saúde	8
12	Conhecimento impreciso: staff	9
13	Exemplo da utente Cristina Ferreira com o demo	9
14	Conhecimento interdito: utente	9
15	Conhecimento interdito: centro de saúde	10
16	Conhecimento interdito: staff	10
17	Conhecimento interdito: vacinação	10
18	Conhecimento interdito, demo	11
19	Registar utente 1 (incerto)	11
20	Registar utente 2 (incerto)	11
21	Registar centro de saúde (incerto)	11
22	Registar staff (incerto)	11
23	Registar vacinação (incerto)	12
24	Registar utente 1 (impreciso)	12
25	Registar utente 2 (impreciso)	12
26	Registar centro (impreciso)	12
27	Registar staff (impreciso)	12
28	Registar utente (interdito)	13
29	Registar centro (interdito)	13
30	Registar staff (interdito)	13
31	Registar vacinação (interdito)	13

1 Introdução

Para esta 2.^a fase, foi-nos proposto, na Unidade Curricular de Sistemas de Representação de Conhecimento e Raciocínio, que estendêssemos o sistema desenvolvido na 1.^a fase deste instrumento de avaliação em grupo, que caracterizava um universo de discurso na área da vacinação global da população portuguesa no contexto COVID.

A elaboração deste instrumento nesta fase 2 foi feita de modo a respeitar as necessidades de demonstração das seguintes funcionalidades:

- Representar conhecimento positivo e negativo;
- Representar casos de conhecimento imperfeito, pela utilização de valores nulos de todos os tipos estudados;
- Manipular invariantes que designem restrições à inserção e à remoção de conhecimento do sistema;
- Lidar com a problemática da evolução do conhecimento, criando os procedimentos adequados;
- Desenvolver um sistema de inferência capaz de implementar os mecanismos de raciocínio inerentes a estes sistemas.

2 Preliminares

Devido ao facto de nesta fase do trabalho ser pedida a extensão da programação em lógica o Pressuposto do Mundo Fechado deve ser abandonado, isto é, a partir de agora deixará de existir apenas conhecimento verdadeiro e falso, pois vamos considerar também a presença do desconhecido.

Com isto surgem dois novos conceitos, nomeadamente a negação forte e conhecimento imperfeito, que poderá ser representado de três formas diferentes:

- **Incerto** - conhecimento acerca do qual não possuímos qualquer informação.
- **Impreciso** - conhecimento acerca do qual sabemos um conjunto/intervalo de valores.
- **Interdito** - conhecimento que não deve ser representado nem conhecido.

3 Descrição do Trabalho e Análise de Resultados

3.1 Conhecimento negativo

Na primeira fase do trabalho, o foco esteve em representar o conhecimento positivo necessário. Com isto, a representação do conhecimento negativo será feita nesta fase. A representação de conhecimento negativo é realizada com base na negação forte, na qual um facto é considerado falso caso não esteja presente na base de conhecimento, nem exista uma exceção a si associada, ou seja, uma certa informação é falsa, caso não seja verdadeira nem desconhecida.

Para o que foi referido, foram utilizados os predicados *nao*, que representa a negação por falha na prova e o *excecao*, que indica a presença de conhecimento imperfeito.

```
% Negação forte do predicado vacinação_Covid  
  
-vacinacao_Covid(S,U,D,V,T) :- nao(vacinacao_Covid(S,U,D,V,T)),  
                                nao(excecao(vacinacao_Covid(S,U,D,V,T))).
```

Figure 1: Exemplo de conhecimento negativo

```
% Extensao do meta-predicado nao: Questao -> {V,F}

nao( Questao ) :- Questao, !, fail.
nao( _ ).
```

Figure 2: Predicado **nao**

3.2 Conhecimento imperfeito

É importante saber-se o tipo de conhecimento representado por uma questão, por isso recorreremos a um predicado *demo* que indica se se trata de conhecimento verdadeiro, falso ou desconhecido.

```
% Extensao do meta-predicado demo: Questao,Resposta -> {V,F}
%                               Resposta = { verdadeiro,falso,desconhecido }

demo( Questao,verdadeiro ) :- Questao.
demo( Questao,falso ) :- -Questao.
demo( Questao,desconhecido ) :- nao( Questao ), nao( -Questao ).
```

Figure 3: Predicado **demo**

```
?- demo(vacinacao_Covid(4,1,(03,02,2021),'Pfizer',1), V).
V = desconhecido.
```

Figure 4: Exemplo de utilização do *demo*

3.2.1 Tipo incerto

Neste tipo de conhecimento, não se tem qualquer informação acerca da informação desconhecida. Sendo assim, este parâmetro desconhecido será caracterizado por um valor nulo, que no caso do utente Fábio Freitas é *profissao-incerta*.

Em conjunto com isto, também é adicionada uma exceção para que o conhecimento que está realcionado com este valor nulo seja representado como desconhecido.

```
utente(9,nss_incerto,'Cláudio Coelho',(09,03,2000),'cc@gmail.com',253444343,'Braga','Motorista',[],2).
```

Figure 5: Conhecimento incerto: utente

```
centro_saude(7,'Centro de saúde da PVZ',morada_incerto,253994854,'cspvz@gmail.com').
```

Figure 6: Conhecimento incerto: centro de saúde

```
staff(9,1,nome_incerto,'jr@gmail.com').
```

Figure 7: Conhecimento incerto: staff

```
vacinacao_Covid(4,1,data_incerto,'Pfizer',1).
```

Figure 8: Conhecimento incerto: vacinação

Para verificar a incerteza, perguntamos ao *demo* e temos um *output* como se mostra na figura abaixo:

```
?- demo(staff(9,1,'Joaquim Almeida','jr@gmail.com'),V).  
V = desconhecido.
```

Figure 9: Exemplo de conhecimento incerto do staff

3.2.2 Tipo impreciso

O tipo impreciso caracteriza-se por representar conhecimento impreciso, que tal como o próprio nome indica, é desconhecido dentro de um conjunto de valores. Os casos que se enquadram nesse intervalo/conjunto de valores são representados como exceções, enquanto que os valores fora são considerados falsos, assim como no exemplo da utente Cristina Ferreira em que o seu número de telefone está entre dois valores.

```
execcao(utente(11,252787999,'Cristina Ferreira',(12,02,1984),'cf@gmail.com',Impreciso,'Lisboa','Apresentadora',[],2)) :-  
    Impreciso >= 253777777, Impreciso =< 253888888.
```

Figure 10: Conhecimento impreciso: utente

```
execcao(centro_saude(8,'Centro de saúde de Beja','Beja',Impreciso,'csbeja@gmail.com')) :-  
    Impreciso >= 253000000, Impreciso =< 253999999.
```

Figure 11: Conhecimento impreciso: centro de saúde

```
excecao(staff(10,Impreciso,'Clara Matias','cm@gmail.com')) :-
    Impreciso >= 1, Impreciso <= 3.
```

Figure 12: Conhecimento impreciso: staff

Exatamente como para o conhecimento do tipo incerto, podemos utilizar o demo para descobrir o tipo de conhecimento impreciso, neste caso, com o exemplo da utente Cristina Ferreira, podemos verificar que dá falso se os valores do seu contacto telefónico estiverem fora do conjunto, e desconhecido se estiverem no conjunto, daí a imprecisão.

```
?- demo(utente(11,252787999,'Cristina Ferreira',(12,02,1984),'cf@gmail.com',253877777,'Lisboa','Apresentadora',[],2),V).
V = desconhecido.

?- demo(utente(11,252787999,'Cristina Ferreira',(12,02,1984),'cf@gmail.com',253977777,'Lisboa','Apresentadora',[],2),V).
V = falso
```

Figure 13: Exemplo da utente Cristina Ferreira com o demo

3.2.3 Tipo interdito

Por fim, o conhecimento imperfeito interdito não deve ser especificado nem conhecido. Para representar este bloqueio atribui-se um valor nulo ao parâmetro pretendido. Tal como os outros tipos, este também é desconhecido, ou seja, é necessário atribuir-lhe uma exceção, tal como o exemplo do utente Ismael Lopes que tem a morada interdita.

```
utente( 13,447565666,'Ismael Lopes',(04,11,1998),'il@gmail.com',253997815,morada_interdito,'Economista',[],1 ).
excecao( utente( Id,Nss,N,D,E,Tel,_,P,Dc,Cs ) ) :-
    utente( Id,Nss,N,D,E,Tel,morada_interdito,P,Dc,Cs ).

nulo( morada_interdito ).
```

Figure 14: Conhecimento interdito: utente

```
centro_saude(9,'Centro de saúde do Porto','Porto',253747252,email_interdito).
execcao( centro_saude(Id,N,M,Tel,_) ) :-
    centro_saude(Id,N,M,Tel,email_interdito).

nulo( email_interdito ).
```

Figure 15: Conhecimento interdito: centro de saúde

```
staff(11,3,'Jorge Jerónimo',email_interdito).
execcao( staff(Ids,Idc,N,_) ) :-
    staff(Ids,Idc,N,email_interdito).

nulo( email_interdito ).
```

Figure 16: Conhecimento interdito: staff

```
vacinacao_Covid(staff_interdito,6,(01,05,2021),'Astrazeneca',1).
execcao( vaccinacao_Covid(_,U,D,V,T) ) :-
    vaccinacao_Covid(staff_interdito,U,D,V,T).

nulo( staff_interdito ).
```

Figure 17: Conhecimento interdito: vacinação

Testando no demo, e com o exemplo da vacinação, obtemos o seguinte:

```
?- demo(vacinacao_Covid(1,6,(01,05,2021),'Astrazeneca',1),V).  
V = desconhecido.  
  
?- evolucao(vacinacao_Covid(1,6,(01,05,2021),'Astrazeneca',1)).  
false.
```

Figure 18: Conhecimento interdito, demo

3.3 Evolução do conhecimento

3.3.1 Conhecimento incerto

```
registraUtente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs,Valor,Incerto) :-  
    Valor == incerto,  
    Incerto == nss,  
    evolucao(utente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs)),  
    inserir((excecao(utente(I,_,Nm,Dt,Em,Tl,Mr,Pf,Dcr,Csd)) :-  
        utente(I,Nss,Nm,Dt,Em,Tl,Mr,Pf,Dcr,Csd))).
```

Figure 19: Registrar utente 1 (incerto)

```
registraUtente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs,Valor,Incerto) :-  
    Valor == incerto,  
    Incerto == profissao,  
    evolucao(utente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs)),  
    inserir((excecao(utente(I,Ns,Nm,Dt,Em,Tl,Mr,_,Dcr,Csd)) :-  
        utente(I,Ns,Nm,Dt,Em,Tl,Mr,Prof,Dcr,Csd))).
```

Figure 20: Registrar utente 2 (incerto)

```
registraCentro(Id,Nome,Mor,Tel,Email,Valor,Incerto) :-  
    Valor == incerto,  
    Incerto == morada,  
    evolucao(centro_saude(Id,Nome,Mor,Tel,Email)),  
    inserir((excecao(centro_saude(I,N,_,T,E)) :-  
        centro_saude(I,N,Mor,T,E))).
```

Figure 21: Registrar centro de saúde (incerto)

```
registraStaff(Id,Idcentro,Nome,Email,Valor,Incerto) :-  
    Valor == incerto,  
    Incerto == nome,  
    evolucao(staff(Id,Idcentro,Nome,Email)),  
    inserir((excecao(staff(I,Ic,_,E)) :-  
        staff(I,Ic,Nome,E))).
```

Figure 22: Registrar staff (incerto)

```
registarVacinao(Idstaff,Idutente,Data,Vac,Toma,Valor,Incerto) :-
    Valor == incerto,
    Incerto == data,
    evolucao(vacinao_Covid(Idstaff,Idutente,Data,Vac,Toma)),
    inserir((excecao(vacinao_Covid(Ids,Idu,_,V,T)) :-
        vacinao_Covid(Ids,Idu,Data,V,T))).
```

Figure 23: Registrar vacinação (incerto)

3.3.2 Conhecimento impreciso

```
registarUtente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs,Valor,Impreciso,Inicio,Fim) :-
    Valor == impreciso,
    Impreciso == telefone,
    inserir((excecao(utente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs)) :-
        Telefone >= Inicio,
        Telefone <= Fim))).
```

Figure 24: Registrar utente 1 (impreciso)

```
registarUtente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs,Valor,Impreciso,Inicio,Fim) :-
    Valor == impreciso,
    Impreciso == centro,
    inserir((excecao(utente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs)) :-
        Cs >= Inicio,
        Cs <= Fim))).
```

Figure 25: Registrar utente 2 (impreciso)

```
registarCentro(Id,Nome,Mor,Tel,Email,Valor,Impreciso,Inicio,Fim) :-
    Valor == impreciso,
    Impreciso == telefone,
    inserir((excecao(centro_saude(Id,Nome,Mor,Tel,Email)) :-
        Tel >= Inicio,
        Tel <= Fim))).
```

Figure 26: Registrar centro (impreciso)

```
registarCentro(Id,Idcentro,Nome,Email,Valor,Impreciso,Inicio,Fim) :-
    Valor == impreciso,
    Impreciso == centro,
    inserir((excecao(staff(Id,Idcentro,Nome,Email)) :-
        Idcentro >= Inicio,
        Idcentro <= Fim))).
```

Figure 27: Registrar staff (impreciso)

3.3.3 Conhecimento interdito

```
registraUtente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs,Valor,Interdito) :-  
    Valor == interdito,  
    Interdito == morada,  
    evolucao(utente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs)),  
    inserir((excecao(utente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs)) :-  
        utente(Id,Nss,Nome,Data,Email,Tel,Mor,Prof,Dc,Cs))),  
    inserir(nulo(Mor)).
```

Figure 28: Registrar utente (interdito)

```
registraCentro(Id,Nome,Mor,Tel,Email,Valor,Interdito) :-  
    Valor == interdito,  
    Interdito == email,  
    evolucao(centro_saude(Id,Nome,Mor,Tel,Email)),  
    inserir((excecao(centro_saude(Id,Nome,Mor,Tel,Email)) :-  
        centro_saude(Id,Nome,Mor,Tel,Email))),  
    inserir(nulo(Email)).
```

Figure 29: Registrar centro (interdito)

```
registraStaff(Id,Idcentro,Nome,Email,Valor,Interdito) :-  
    Valor == interdito,  
    Interdito == email,  
    evolucao(staff(Id,Idcentro,Nome,Email)),  
    inserir((excecao(staff(Id,Idcentro,Nome,Email)) :-  
        staff(Id,Idcentro,Nome,Email))),  
    inserir(nulo(Email)).
```

Figure 30: Registrar staff (interdito)

```
registraVacinao(Idstaff,Idutente,Data,Vac,Toma,Valor,Interdito) :-  
    Valor == interdito,  
    Interdito == staff,  
    evolucao(vacinacao_Covid(Idstaff,Idutente,Data,Vac,Toma)),  
    inserir((excecao(vacinacao_Covid(Idstaff,Idutente,Data,Vac,Toma)) :-  
        vacinacao_Covid(Idstaff,Idutente,Data,Vac,Toma))).
```

Figure 31: Registrar vacinação (interdito)

4 Conclusão

A realização deste trabalho foi fundamental, pois foi uma forma de consolidarmos o conteúdo abordado, relativo à programação em lógica e consideramos que as funcionalidades foram bem implementadas, na medida em que todos os requisitos foram cumpridos e fizemos jus ao que foi desenvolvido na primeira fase, dando-lhe um bom seguimento.

Para isso, para além do que nos foi pedido nesta fase corrigimos certos aspetos, adicionamos invariantes e incluímos a remoção de conhecimento nas funcionalidades disponíveis.