**Universidade do Minho**

**2º Exercício**

Mestrado Integrado em Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio

(2º Semestre / 2016-2017)

A74859 João da Cunha Coelho

A74601 José Miguel Ribeiro da Silva

A73959 Pedro João Novais da Cunha

A74748 Luís Miguel Moreira Fernandes

Braga,

Março de 2017

**Resumo**

O presente documento tem como objetivo a documentação do segundo exercício prático da unidade curricular de *Sistemas de Representação de Conhecimento e Raciocínio*. Seguindo o modelo do primeiro exercício prático, após uma breve introdução, é detalhada a forma como o grupo de trabalho desenvolveu as várias funcionalidades propostas no enunciado do problema. Após a apresentação de alguns resultados, é realizada uma breve crítica ao trabalho realizado.

**Índice**

**Introdução**

O segundo exercício prático tem como objetivo a elaboração de um sistema de representação de conhecimento e raciocínio contextualizado numa situação em que a informação existente sobre o problema não é completa, neste caso em específico, a área da prestação de cuidados de saúde.

O exercício prático incide assim sobre a representação de conhecimento imperfeito onde é introduzido um terceiro valor de verdade, o desconhecido. Fazendo uso das características da programação em lógica, em especial, da linguagem PROLOG, fica assim clara a necessidade de desenvolvimento de novos mecanismos capazes de lidar e atuar sobre este novo tipo de conhecimento.

Ao longo do exercício prático para além do desenvolvimento dos mecanismos referidos acima, são explicitados vários predicados exemplificativos das três variantes do conhecimento imperfeito, bem como invariantes capazes de restringir a inserção e remoção destes.

**Preliminares**

A realização do primeiro exercício prático da unidade curricular permitiu uma maior facilidade no que diz respeito à manipulação da linguagem utilizada para a realização deste segundo trabalho, na construção de invariantes e predicados exemplificativos dos vários tipos de conhecimento. Por outro lado, a presença por parte de todos os elementos do grupo nas aulas teóricas facilitou sobretudo a implementação de mecanismos capazes de lidar com o novo tipo de conhecimento.

Com um papel menos relevante, breves pesquisas na web, a leitura prévia e atenta do enunciado, a revisão da matéria e a distribuição de tarefas pelos diferentes membros do grupo, permitiram igualmente que a implementação das funcionalidades propostas se iniciassem de forma tranquila e se prolongasse sem sobressaltos.

**Descrição do trabalho e análise dos resultados**

* **Inserção de conhecimento**

À semelhança do primeiro exercício prático, o objetivo foi desenvolver um sistema de representação de raciocínio sobre a prestação de cuidados de saúde pela realização de serviços de atos médicos. Para além disso, neste caso em particular, consideramos que a informação existente sobre o problema não é completa. Como consequência, aos valores de verdades já contemplados anteriormente, verdadeiro ou falso, foi introduzido um novo, o desconhecido. Deste modo, as extensões de predicado realizadas no exercício anterior tiveram que ser ligeiramente alteradas. Apesar dos exemplos a seguir apresentados incidirem sobre conhecimento perfeito, mais à frente, no relatório, vai ser exemplificada a inserção de conhecimento imperfeito que justifica a inserção do novo valor de verdade. Assim, no que diz respeito ao conhecimento perfeito, temos:

% Extensão do predicado utente: IdUt, Nome, Idade, Sexo, Morada -> {V, F, D}

**utente**( 1, joao, 20, masculino, 'vila verde' ).

**utente**( 2, jose, 20, masculino, 'lousada' ).

**utente**( 3, josefina, 34, feminino, 'aveiro' ).

**utente**( 4, luis, 20, masculino, 'vila das aves' ).

**utente**( 5, pedro, 20, masculino, 'felgueiras' ).

% Extensão do predicado cuidadoPrestado: IdServ, Descrição, Instituição, Cidade -> {V, F, D}

**cuidadoPrestado**( 1, 'Medicina Familiar', 'Centro de Saude de Vila Verde', 'Vila Verde').

**cuidadoPrestado**( 2, 'Radiologia', 'Hospital Sao Joao', 'Porto').

**cuidadoPrestado**( 3, 'Medicina Familiar', 'Centro de Saude de Lousada', 'Lousada').

**cuidadoPrestado**( 4, 'Medicina Familiar', 'Centro de Saude de Felgueiras', 'Felgueiras').

**cuidadoPrestado**( 5, 'Ginecologia', 'Hospital de Braga', 'Braga').

**cuidadoPrestado**( 6, 'Obstetricia', 'Hospital de Braga', 'Braga').

% Extensão do predicado atoMedico: Data, IdUt, IdServ, Custo -> {V, F, D}

**atoMedico**( '14-03-2017', 1, 5, 30 ).

**atoMedico**( '12-03-2017', 3, 2, 20 ).

**atoMedico**( '13-03-2017', 4, 4, 5 ).

**atoMedico**( '14-03-2017', 2, 3, 5 ).

**atoMedico**( '04-04-2017', 1, 3, 7 ).

Os exemplos práticos apresentados em cima referem-se a conhecimento perfeito positivo e em nada diferem do que foi realizado no exercício prático anterior. No entanto, para o presente exercício prático um dos objetivos passou por ser igualmente possível a inserção conhecimento perfeito negativo. Assim, foi inserido conhecimento perfeito negativo para os três principais predicados, como podemos verificar a seguir:

**-utente**( 17, laura, 24, feminino, 'Faro' ).

**-utente**( 34, ricardo, 38, masculino, 'Guarda' ).

**-Utente**( 72, alberto, 83, masculino, 'Setubal' ).

Com os predicados explicitados acima é possível representar que determinado utente não se encontra presente na base de conhecimento. O mesmo pode ser assumido para os seguintes predicados, correspondentes aos cuidados prestados e atos médicos, respetivamente:

**-cuidadoPrestado**( 30, 'Medicina Familiar', 'Hospital de São João', 'Porto').

**-cuidadoPrestado**( 45, 'Obstetrícia', 'Hospital do Algarve', 'Porto').

**-cuidadoPrestado**( 98, 'Ginecologia', 'Centro de Saude de Felgueiras', 'Felgueiras').

**-atoMedico**( '31-01-2017', 5, 3, 13 ).

**-atoMedico**( '12-07-2017', 3, 4, 43 ).

**-atoMedico**( '06-01-2018', 2, 1, 59 ).

**-atoMedico**( '24-12-2017', 6, 5, 2 ).

Tal como no primeiro exercício prático, é necessário controlar a inserção e remoção de utentes, cuidados prestados e atos médicos. A inserção de conhecimento perfeito negativo levou inevitavelmente à criação de novos invariantes estruturais e referenciais de inserção e remoção. Mas antes é importante expor os predicados auxiliares utilizados e o que se alterou relativamente à primeira fase.

**Explicar de seguida predicados auxiliares novos**

**Explicar nova evolução…**

No que diz respeito aos invariantes estruturais, se no exercício passado foi realizado um invariante que garantia, na inserção de conhecimento perfeito positivo, a unicidade dos Ids dos utentes, o mesmo tem que ser realizado para a inserção de conhecimento perfeito negativo. Isto não é apenas verificado para o predicado utente mas também para os restantes predicados utilizados no exercício. No que diz respeito aos cuidados prestados garantimos a unicidade dos Ids dos serviços e no que diz respeito aos atos médicos…**DÚVIDA**…Assim, temos:

% Garantia de unicidade nos Ids dos utentes – conhecimento perfeito positivo

+**utente**( IdUt,Nome,Idade,Sexo,Morada )::(**solucoes**( (IdUt), **utente**(IdUt,N,I,Se,M), S ), **comprimento**( S,N ), N == 0 ).

% Garantia de unicidade nos Ids dos utentes – conhecimento perfeito negativo

---------------------------------------------------------------

**Unicidade nos Ids dos utentes – conhecimento negativo - utente**

**---------------------------------------------------------------**

% Garantia de unicidade nos Ids dos Serviços

+cuidadoPrestado(IdServ,Desc,Inst,Cid) :: (solucoes( (IdServ),cuidadoPrestado(IdServ,D,I,C),S), comprimento(S,N), N == 0).

% Garantia de unicidade nos Ids dos Serviços

+(-cuidadoPrestado(IdServ,Desc,Inst,Cid)) :: (solucoes( (IdServ),-cuidadoPrestado(IdServ,D,I,C),S), comprimento(S,N ), N == 0).

---------------------------------------------------------------

**Unicidade atos médicos**

**---------------------------------------------------------------**

A possibilidade de inserção de conhecimento perfeito negativo trouxe igualmente a necessidade de garantir que não existe conhecimento contraditório, ou seja, não é possível a inserção de conhecimento perfeito negativo se na base de conhecimento já existir conhecimento perfeito positivo exatamente com a mesma informação. O mesmo se verifica para quando existe conhecimento perfeito negativo e queremos inserir conhecimento perfeito positivo com a mesma informação. Mais uma vez, isto não é apenas verificado para o predicado utente mas também para os restantes predicados utilizados no exercício. Assim, temos:

% Garante que não existe conhecimento positivo contraditótio

+(-utente(IdUt,Nome,Idade,Sexo,Cidade)) :: (solucoes((IdUt), utente(IdUt,Nome,Idade,Sexo,Cidade), S ), comprimento( S,N ), N == 0).

---------------------------------------------------------------

**Falta conhecimento negativo contraditório - utente**

**---------------------------------------------------------------**

% Garantir que não existe conhecimento negativo contraditótio

+cuidadoPrestado(IdServ,Desc,Inst,Cid) :: (solucoes((IdServ), -cuidadoPrestado(IdServ,Desc,Inst,Cid), S ), comprimento( S, N ), N ==0).

% Garantir que não existe conhecimento positivo contraditótio

+(-cuidadoPrestado(IdServ,Desc,Inst,Cid)) :: (solucoes( (IdServ),cuidadoPrestado(IdServ,Desc,Inst,Cid), S ),comprimento( S, N ), N == 0 ).

% Garantir que não existe conhecimento positivo contraditótio

+(-atoMedico(Data,IdUt,IdServ,Custo)) :: (solucoes( (IdUt,IdServ),atoMedico(Data,IdUt,IdServ,Custo),S), comprimento( S, N ),N == 0 ).

---------------------------------------------------------------

**Falta conhecimento negativo contraditório – ato médico**

**---------------------------------------------------------------**

* **Inserção de conhecimento imperfeito**

Como já foi dito anteriormente, o presente trabalho prático não assenta apenas sobre conhecimento perfeito. Através o novo valor de verdade é agora possível a inserção de conhecimento incerto, impreciso e interdito.

No que diz respeito ao **conhecimento imperfeito do tipo interdito** este ocorre quando determinado facto não se conhece e permanecerá sempre uma incógnita. Seguem dois exemplos:

% Não se sabe nem é possível vir a saber a morada do utente % António Costa.

utente(11, 'Antonio Costa', 55, masculino, int0001).

excecao(utente(Id,Nome,Idade,Sexo,Morada)) :- utente( Id,Nome,Idade,Sexo,int0001).

nulo(int0001).

% Não se sabe nem é possível vir a saber qual o serviço prestado ao utente cujo % IdUt é 10 no dia 20 de Março de 2017 e cujo preço foi 3000€.

atoMedico( '20-03-2017', 11, int0002, 3000 ).

excecao(atoMedico(Data,IdUt,IdServ,Custo)) :- atoMedico( Data,IdUt,int0002,Custo ).

nulo(int0002).

Como podemos verificar por os exemplos acima, os valores desconhecidos são previamente definidos como sendo valores nulos. Tomando como exemplo o primeiro predicado, para o caso do utente ‘António Costa’ a morada é desconhecida, sendo definida em int0001, como um valor nulo. Para além disto, é necessário garantir que não é possível alterar o valor nulo refiro anteriormente através de uma inserção. Apesar de já possuirmos um invariante que garante a unicidade dos Ids para os utentes e assim impede a alteração do valor nulo, decidimos, seguindo a metodologia proposta nas aulas práticas, criar invariantes especialmente realizados para cobrir este caso em específico:

% Invariante que impede a inserção de conhecimento positivo ou negativo acerca de conhecimento interdito sobre a cidade de utentes

+utente(Id,No,I,Se,C) :: (solucoes((Id,No,I,Se,C),(utente( Id,No,I,Se,xpto ),nulo(xpto)),S),comprimento( S,N ),N==0).

+(-utente(Id,No,I,Se,C)):: (solucoes((Id,No,I,Se,C), (utente(Id,No,I,Se,xpto),nulo(xpto)),S),comprimento(S,N),

N == 0).

No que diz respeito ao **conhecimento imperfeito do tipo impreciso** este ocorre quando determinado facto não se conhece mas a dúvida incide sobre um determinado intervalo de incerteza. Seguem vários exemplos:

% Não se sabe se o/a utente Dolores é do sexo masculino ou feminino.

excecao(utente( 7, dolores, 34, masculino, 'Amadora' )).

excecao(utente( 7, dolores, 34, feminino, 'Amadora' )).

% Não se sabe se o utente Zeca tem 36 ou 37 anos, nem se é da Amadora ou de Sintra.

excecao(utente( 8, zeca, 36, masculino, 'Sintra' )).

excecao(utente( 8, zeca, 37, masculino, 'Sintra' )).

excecao(utente( 8, zeca, 36, masculino, 'Amadora' )).

excecao(utente( 8, zeca, 37, masculino, 'Amadora' )).

% Não se sabe se o utente Alfredo é de Felgueiras ou Lousada.

excecao(utente( 9, alfredo, 22, masculino, 'felgueiras' )).

excecao(utente( 9, alfredo, 22, masculino, 'lousada' )).

% Não se sabe se a utente Alzira tem 23 ou 24 anos.

excecao(utente( 10, alzira, 23, feminino, 'braga' )).

excecao(utente( 10, alzira, 24, feminino, 'braga' )).

% Não se sabe o preço da consulta que ocorreu no dia 29 de Abril % de 2017, cujo utente tem o IdUt 2 e o serviço prestado tem o % IdServ 3, mas sabe-se que o preço foi entre os 3 e os 17 euros.

excecao(atoMedico('29-04-2017',2,3,C)) :- C>=3, C=<17.

Nos casos anteriormente explicitados, com a utilização das exceções garantimos que o sistema irá reconhecer determinada questão como sendo desconhecida. Utilizando o pressuposto do mundo fechado, garantimos que caso contrário a resposta devolvida é falsa. De seguida segue o exemplo do pressuposto do mundo fechado para o predicado utente:

% Admissão do pressuposto do mundo fechado para o predicado utente

-utente(Id,N,I,S,C) :-

nao(utente(Id,N,I,S,C)),

nao(excecao(utente(Id,N,I,S,C))).

**Conclusões e Sugestões**

**Referências**

* “PROLOG: Programming for Artificial Intelligence”, Ivan Bratko;
* “Representação de Informação Incompleta”, Cesar Analide, José Neves;
* “Sugestões para a Redacção de Relatórios Técnicos, Cesar Analide, Paulo Novais, José Neves;

**Anexos**