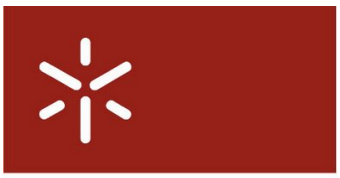


Universidade do Minho

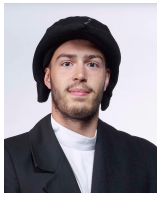
Sistemas de Representação de Conhecimento e Raciocínio

MIEI - 3º ano - 2º semestre
Universidade do Minho

Trabalho II
Programação em lógica estendida e Conhecimento Imperfeito

Grupo 10


Dinis Peixoto
A75353


José Bastos
A74696


Ricardo Pereira
A74185


Marcelo Lima
A75210

16 de Novembro de 2017

1. *Resumo*

O presente documento corresponde ao relatório elaborado em conjunto com a resolução do segundo exercício da componente prática exigida na unidade curricular *Sistemas de Representação de Conhecimento e Raciocínio*. Com o tema *Programação em lógica estendida e Conhecimento imperfeito* e sendo este a continuação do trabalho anteriormente elaborado, a resolução do exercício pretende continuar a motivar os alunos para a linguagem de programação em lógica *PROLOG* à medida que esta vai sendo aprofundada.

Assim, é exigido o desenvolvimento de um sistema de representação de conhecimento e raciocínio com capacidade para caracterizar um universo de discurso na área da prestação de cuidados de saúde pela realização de serviços e atos médicos a utentes de diferentes estabelecimentos, tal como na fase anterior. No entanto, nesta fase temos de ser capazes de demonstrar as funcionalidades subjacentes à programação em lógica estendida e à representação de conhecimento imperfeito, recorrendo à temática dos valores nulos.

Desta forma, várias das funcionalidades implementadas na fase anterior continuarão inalteradas e algumas destas serão modificadas de modo a cumprir com os requisitos necessários.

Conteúdo

2. *Introdução*

Este relatório é o resultado da elaboração do segundo exercício prático proposto na unidade curricular de *Sistemas de Representação de Conhecimento e Raciocínio*, com o tema *Programação em lógica estendida e Conhecimento imperfeito*.

Uma vez que este exercício se trata da continuação do trabalho elaborado, este continuará a ter por base a utilização da programação em lógica, mais concretamente da linguagem de programação *PROLOG*. No entanto, desta vez o objetivo é outro, o qual passa por promover a representação de conhecimento imperfeito, recorrendo à utilização de valores nulos.

Tal como no exercício anterior, no enunciado é-nos pedido que seja desenvolvido um sistema de representação de conhecimento e raciocínio com capacidade para caracterizar todo um vasto cenário centralizado na área da prestação de cuidados de saúde, mais propriamente na realização de serviços de atos médicos. Assim, existem três principais fontes de conhecimento: os utentes, os cuidados prestados e os atos médicos. O objetivo é criar um sistema capaz de receber e guardar todas as informações associadas a cada um destes e ao mesmo tempo conseguir lidar com certas restrições de conhecimento, restrições estas que no seu conjunto fazem parte daquilo que é chamado conhecimento imperfeito.

Desta forma, nas próximas secções serão abordados os vários tipos de conhecimento imperfeito, explicando em que é que consistem e como é que estes serão aplicados ao nosso cenário em particular.

3. *Preliminares*

3.1 Conhecimentos anteriormente assimiliados

Sendo este o segundo exercício proposto na UC, é natural que nesta altura já tenhamos assimilado algum conhecimento aquando da realização do exercício anterior, o qual será igualmente importante para a resolução deste trabalho.

Desta forma, com o desenvolvimento do anterior sistema de representação de conhecimento e raciocínio, já nos foi possível assimilar de um modo genérico aquilo que é a programação em lógica e o que é que esta tem por base para o seu funcionamento. Ao mesmo tempo, também adquirimos conhecimentos mais concretos no que toca a esta temática, como por exemplo, a utilização de invariantes para a remoção e inserção de conhecimento, os quais nos ajudam a manter a integridade da informação dentro do sistema de conhecimento que desejamos criar.

Por outro lado, na transição para este exercício, há algo bastante importante que irá surgir naquilo que é a representação de conhecimento. No exercício anterior, o nosso sistema funcionava como um mundo fechado e portanto as respostas às questões só poderiam ser de dois tipos: verdadeiro ou falso, consoante existisse ou não informação no sistema de conhecimento que provasse a veracidade de tais afirmações. Agora, com este exercício, surge uma nova forma de representar conhecimento, que é a representação de conhecimento imperfeito, ou por outras palavras, informação que é desconhecida. Neste caso, a resposta às questões já podem ser de 3 tipos: verdadeiro, falso ou desconhecido, este último ocorre caso não existam provas que uma afirmação seja de cariz verdadeiro ou falso.

3.2 Bases de Dados ou Representação do Conhecimento

Os sistemas de Bases de Dados (BD's) e os sistemas de representação de conhecimento, lidam com aspectos concretos do mundo real e podem-se comparar nos termos em que dividem a utilização da informação. Ambos os sistemas distinguem as funções de ***representação*** *– descrição do esquema conceptual e dos dados – e de* ***computação*** *– construção de respostas a questões e manipulação dos dados.*

Desta forma, apresentaremos de seguida, os pressupostos pelos quais se baseiam as linguagens de manipulação tanto os sistemas de bases de dados (BD's) como dos sistemas de representação de conhecimento.

3.2.1 Bases de Dados

Os pressupostos em que se baseiam as linguagens de manipulação de base de dados seguem o ideal de que apenas existe e é válida toda a informação contida nesta, sendo que, toda a restante informação é considerada falsa. Apresentaremos, de seguida, os

pressupostos nos quais as linguagens de manipulação de informação num Sistema de Base de Dados se baseiam:

- **Pressuposto do Mundo Fechado** - toda a informação que não existe mencionada na base de dados é considerada falsa;
- **Pressuposto dos Nomes Únicos** - duas constantes diferentes (que definam valores atómicos ou objetos) designam, necessariamente, duas entidades diferentes do universo de discurso;
- **Pressuposto do Domínio Fechado** - não existem mais objectos no universo de discurso para além daqueles designados por constantes na base de dados.

3.2.2 Sistema de Representação de Conhecimento

Porém, os Sistemas de Representação de Conhecimento nem sempre pretendem assumir que a informação representada é a única que se pode considerar válida e que as entidades representadas sejam as únicas existentes no mundo exterior. Seguem, então, os pressupostos considerados para um sistema deste tipo:

- **Pressuposto do Mundo Aberto** - podem existir outros factos ou conclusões verdadeiros para além daqueles representados na base de conhecimento;
- **Pressuposto dos Nomes Únicos** - duas constantes diferentes (que definam valores atómicos ou objectos) designam, necessariamente, duas entidades diferentes do universo de discurso;
- **Pressuposto do Domínio Aberto** - podem existir mais objectos do universo de discurso para além daqueles designados pelas constantes da base de conhecimento.

3.3 Representação de Informação Incompleta

No mundo real somos frequentemente enfrentados com questões onde a informação existente é insuficiente e/ou incompleta. Perante estas situações é importante saber representá-las para além do verdadeiro ou falso, sendo necessário demonstrar que a questão/situação em específico é incompleta/desconhecida e que o seu valor não pode ser definido de imediato, ao contrário do que acontece na maior parte das situações para as mais variadas linguagens de programação com que somos familiarizados ao longo do tempo.

Assim, surge a programação em lógica estendida (PLE) capaz de resolver estas situações. Através desta seremos capazes de representar tais situações de conhecimento desconhecido, ou por outras palavras, de informação incompleta. Representaremos o conhecimento seguindo os três diferentes tipos de conclusões para uma questão:

- **Verdadeiro** - quando for possível provar uma questão(Q) na base de conhecimento;
- **Falso** - quando for possível provar a falsidade de uma questão(-Q) na base de conhecimento;
- **Desconhecido** - quando não for possível provar a questão(Q) nem a questão(-Q) na base de conhecimento.

De facto, para cumprir com os pressupostos definidos anteriormente teremos de definir outro tipo de informação, além da informação positiva e negativa, já anteriormente abordadas no primeiro trabalho prático. Passam então a existir dois tipos de negação diferentes:

- **Negação por Falha** - quando não existe nenhuma prova. É representada pelo teorema *não* utilizado no primeiro trabalho prático;
- **Negação Forte** - quando afirma que um determinado predicado é falso. É representada pelo teorema (-).

3.4 Valores nulos

Na problemática da representação da informação incompleta, os valores nulos são aqueles que surgem com mais frequência e com o objetivo de separar dois tipos de situações, aquelas em que as respostas a questões devem ser dadas como conhecidas (verdadeiras ou falsas) e aquelas, que por outro lado, devem ser dadas como desconhecidas.

Estes valores nulos podem ser divididos em três tipos distintos, cada um com as suas caraterísticas. O primeiro permite representar valores desconhecidos; o segundo permite respresentar valores desconhecidos, mas agora de um conjunto determinado de valores; e por último, um pouco distinto dos anteriores, é aquele que representa valores não permitidos, definidos como interditos na base de conhecimento considerada. A seguir, de maneira a compreendermos melhor o que cada um destes representa, serão expostos com mais detalhe estes três tipos de valores nulos.

3.4.1 Desconhecido

Tal como o nome indica, este tipo de valor nulo serve para representar algo que não é conhecido, ou seja, que não exista nenhuma prova que comprove que isso seja verdadeiro ou falso. Imaginemos, por exemplo, que o João não sabe quem é o seu pai, a resposta ao predicado pai(X,Joao) - quem é o pai do João, nestas circunstâncias, teria de ser desconhecida, no sentido em que a questão é verdadeira, porque tem solução, mas a concretização dessa solução é que é desconhecida, já que não se sabe em concreto quem é o pai do João, apenas se sabe que o João tem obrigatoriamente um pai.

3.4.2 Desconhecido, de um conjunto de valores

A diferença entre o valor nulo do tipo desconhecido, visto anteriormente, e o valor nulo desconhecido, mas de um conjunto determinado de valores, está precisamente no facto de este último valor nulo representar um (ou mais) valores de um conjunto de valores bem determinados, só não é conhecido, especificamente, qual dos valores concretizará a questão.

Assim, qualquer resposta ao predicado que utilize um valor fora desse conjunto resultará em falsidade, enquanto que se este estiver dentro do conjunto, o resultado será desconhecido, uma vez que a informação continua a ser desconhecida, apenas se sabe que o valor está contido nesse mesmo conjunto.

3.4.3 Não permitido

Este último tipo de valores nulos, tal como dito anteriormente, é um pouco diferente do dois tipos anteriores, uma vez que, para além de identificarem, como os anteriores,

valores desconhecidos, caracterizam, ainda, um tipo de dados que não se pretende admitir que surjam na base de conhecimento.

Por exemplo, temos o predicado medicamento, que nos diz o nome do medicamento que cura determinada doença. A resposta à questão medicamento(X,cancro), sendo X um valor nulo não permitido, será de cariz desconhecido, pois, para além de identificar um valor desconhecido, pretende significar que o valor que representa não é permitido especificar ou conhecer, uma vez que não existe cura para o cancro, e qualquer tentativa posterior de concretizar esse valor deverá ser rejeitada como sendo provocadora de inconsistência na informação constante da base de conhecimento.

4. *Descrição do Trabalho e Análise de Resultados*

Tal como já foi anteriormente dito, foi-nos proposta a realização de um trabalho sobre esta temática que é o conhecimento imperfeito, a qual foi esclarecida no capítulo anterior.

Desta forma, neste capítulo abordaremos todos os tópicos que nos foram requisitados no enunciado do trabalho, tentando expô-los da melhor maneira possível, e desta forma explicar como é que conseguimos atingir o resultado final. Os tópicos requisitados foram os seguintes:

- 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.

4.1 Fontes de conhecimento

O universo que se pretende representar mantém-se a realização de serviços de atos médicos e, como tal, as fontes de conhecimento anteriormente representadas também mantêm-se com pequenas alterações.

A representação da *Data* nos *Atos* era pouco intuitiva e exigia elevados conhecimentos de manipulação de *strings* na linguagem de programação em lógica utilizada, posto isto, o grupo optou pela criação de uma nova fonte de conhecimento Data, facilitando todo o processo requerido no manuseio desta. Além disso e uma vez que agora é exigida a representação de **informação incompleta** o domínio de soluções foi alterado passando a incluir também o *Desconhecido*.

Desta forma, segue-se a definição das fontes de conhecimento representadas:

- **utente**: (#ID_Utente, Nome, Idade, Morada) → {V,F,D}
- **serviço**: (#ID_Serviço, Descrição, Instituição, Cidade) → {V,F,D}

- **ato**: (Data, #ID_Utente, #ID_Serviço, Custo) → {V,F,D}
- **medico**: (#ID_Médico, Nome, Idade, Morada, Especialização) → {V,F,D}
- **data**: (Dia, Mês, Ano) → {V,F,D}

Seguem as definições iniciais correspondentes às fontes de conhecimento anteriores.

```
:- dynamic utente/4.        % (ID_Utente, Nome, Idade, Morada)
:- dynamic servico/4.       % (ID_Serviço, Descrição, Instituição, Cidade).
:- dynamic ato/5.           % (Data, ID_Utente, ID_Serviço, Custo, ID_Médico).
:- dynamic medico/5.        % (ID_Médico, Nome, Idade, Morada, Especialização).
:- dynamic data/3.          % (Dia, Mês, Ano)
```

4.2 Representação de conhecimento positivo e negativo

No primeiro trabalho prático realizado a representação de **conhecimento positivo** já tinha sido realizada, esta como base de conhecimento inicial, de forma a que fosse possível testar os diferentes predicados sem ter necessariamente de inserir conhecimento previamente. Ao efetuar o preenchimento da base de conhecimento inicial tivemos cuidado com as definições das fontes de conhecimento.

```
utente(1,'Renato Portoes',32,'Rua Nova Santa Cruz').
utente(2,'Renato Portoes',32,'Rua Nova Santa Cruz').
utente(3,'Ricardo Azevedo',20,'Rua Velha Santa Cruz').
utente(4,'Ana Martins',22,'Rua do Outeiro').
utente(5,'Jose Vilaca',12,'Rua das Cegonhas').
utente(6,'Sara Alberto',45,'Rua dos Marcianos').
utente(7,'Joao Guilherme',89,'Avenida Central').
utente(8,'Afonso Aragao',26,'Rua dos Palacetes').
utente(9,'Jose Silva',73,'Estreito Largo').
utente(10,'Frederico Pereira',4,'Rua dos Carretos').
utente(11,'Rita Gameiro',52,'Rua Engenheiro Antonio Filipe').
utente(12,'Luis Marcio',15,'República das Bananas').
utente(13,'Pedro Luis',53,'Largo do Bigode').
utente(14,'Marco Cardoso',33,'Avenida Principal').


servico(1,'Ortopedia','Centro de Saude Santa Tecla','Braga').
servico(2,'Cardiologia','Centro de Saude Santa Tecla','Braga').
servico(3,'Neurocirurgia','Hospital Privado de Braga','Braga').
servico(4,'Pediatria','Posto medico de Fao','Esposende').
servico(5,'Otorrinolaringologia','Hospital Santa Maria','Porto').
servico(6,'Oftalmologia','Clinica de Santa Madalena','Felgueiras').
servico(7,'Urologia','Casa de Saude de Caldelas','Amares').
servico(8,'Ginecologia','Clinica do Tubarao','Viana do Castelo').
servico(9,'Ortopedia','Espaco Saude Beirao Rendeiro','Moledo').


medico(1,'Dr. Eugenio Andrade',54,'Viana do Castelo','Ortopedia').
medico(2,'Dr. Firmino Cunha',63,'Guimaraes','Cardiologia').
medico(3,'Dr. Jorge Costa',38,'Santo Tirso','Neurocirurgia').
medico(4,'Dr. Bruno Hermenegildo',48,'Vila das Aves','Pediatria').
medico(5,'Dra. Alberta Mendes',57,'Matosinhos','Otorrinolaringologia').
medico(6,'Dr. Cristiano Soares',34,'Terras de Bouro','Oftalmologia').
medico(7,'Dra. Rosalina Oliveira',49,'Vieira do Minho','Urologia').
medico(8,'Dr. Carlos Mota',44,'Maia','Ginecologia').


ato('12-01-2017',1,1,19,1).
ato('13-01-2017',14,2,10,2).
ato('14-01-2017',2,3,15,3).
ato('15-01-2017',13,4,5,4).
ato('16-01-2017',3,6,12,6).
ato('17-01-2017',12,6,14,6).
ato('15-01-2017',4,7,5,7).
ato('16-01-2017',11,5,12,5).
ato('17-01-2017',5,8,100,8).
ato('24-01-2017',6,2,35,2).
ato('15-01-2017',7,7,80,7).
ato('19-01-2017',8,2,200,2).
ato('11-01-2017',9,6,10,6).
ato('25-01-2017',10,2,55,2).
ato('26-01-2017',11,3,40,3).
ato('29-01-2017',12,4,90,4).
ato('31-01-2017',13,9,50,1).
```

Contudo, na realização do presente trabalho prático foi necessário adicionar também **conhecimento negativo**, sendo que a importância deste na PLE já foi anteriormente explicada. O conhecimento negativo foi representado de dois diferentes tipos: **negação por falha** e **negação forte**.

```
-utente(ID_Utente, Nome, Idade, Morada) :- nao(utente(ID_Utente, Nome, Idade, Morada)),
                                           nao(excecao(utente(ID_Utente, Nome, Idade, Morada))).

-servico(ID_Servico, Descricao, Instituicao, Cidade) :- nao(servico(ID_Servico, Descricao, Instituicao, Cidade)),
                                                       nao(excecao(servico(ID_Servico, Descricao, Instituicao, Cidade))).

-ato(Data, ID_Utente, ID_Servico, Custo, ID_Medico) :- nao(ato(Data, ID_Utente, ID_Servico, Custo, ID_Medico)),
                                                       nao(excecao(ato(Data, ID_Utente, ID_Servico, Custo, ID_Medico))).

-medico(ID_Medico, Nome, Idade, Morada, Especializacao) :- nao(medico(ID_Medico, Nome, Idade, Morada, Especializacao)),
                                                           nao(excecao(medico(ID_Medico, Nome, Idade, Morada, Especializacao))).

% Exemplos de conhecimento negativo.
-utente(25,'Antonio Manuel',47,'Avenida Manuel Carvalho').
-servico(15,'Cardiologia','Clinica Limao','Guimaraes')
-ato(data(20,05,2017),1,7,87,7).
-medico(9,'Dr. Fernando Silva',36,'Porto','Ginecologia').
```

4.3 Representação de casos de conhecimento imperfeito

A utilização de valores nulos surgem como uma estratégia adoptada para a distinção de tipos de situações, isto é, quando se pretende fazer a distinção entre situações em que as respostas às questões são concretizadas como conhecidas (verdadeiras ou falsas) ou quando estas são simplesmente desconhecidas.

Existem, portanto, três tipos de conhecimento imperfeito: conhecimento incerto, conhecimento impreciso e ainda, conhecimento interdito. Explicaremos então a representação de cada um destes tipos de conhecimento imperfeito.

4.3.1 Conhecimento Incerto

Quando se trata de um valor nulo do tipo desconhecido e não necessariamente de um conjunto determinado de valores.

```
% Desconhecimento da morada do utente.
utente(15,'Antonio Manuel',47,morada_desconhecida).
utente(16,'Eduardo Fidalgo',11,morada_desconhecida).

% Desconhecimento da idade do utente, mas com o conhecimento de que não é 50 anos.
utente(17,'Ze Maria',idade_desconhecida,'Rua das azeitonas').
-utente(17,'Ze Maria',50,'Rua das azeitonas').

% Conjunto das exceções associadas.
excecao(utente(ID_Utente, Nome, Idade, Morada)) :- utente(ID_Utente, Nome, Idade,morada_desconhecida).
excecao(utente(ID_Utente, Nome, Idade, Morada)) :- utente(ID_Utente, Nome, idade_desconhecida, Morada).
```

4.3.2 Conhecimento Impreciso

A diferença entre este e o anterir é que, nesta situação, o valor nulo representará um ou mais valores de um conjunto de valores bem determinado, não sendo conhecido especificamente, qual o valor do conjunto considerado que concretizará a questão.

```
% A Dra. Ana Fonseca abriu recentemente uma clínica que oferece serviços de uma só
% especialidade. Sabe-se também que a doutora se especializou em Cardiologia,
% Neurologia e ainda Neurocirurgia, desta forma, o serviço oferecido pela clínica
% pode efetivamente ser de qualquer uma das especialidade da doutora.

excecao(servico(12,'Cardiologia','Clinica Fonseca','Guimaraes')).
excecao(servico(12,'Neurologia','Clinica Fonseca','Guimaraes')).
excecao(servico(12,'Neurocirurgia','Clinica Fonseca','Guimaraes')).

% Devido a um problema na base de dados do Hospital, foi perdido o dia referente
% a um determinado ato, sabendo-se agora apenas o ano e mês deste.

excecao(ato(data(Dia,03,2017),6,3,85,3)) :- Dia>=1, Dia=<31.

% Devido a um problema na base de dados do Posto Médico, foi perdido o mês referente
% a um determinado ato, sabendo-se agora apenas o dia e o ano deste.

excecao(ato(data(5,Mes,2017),2,4,40,4)) :- Mes>=1, Mes=<12.

% O António contou ao José que na sua última consulta de Oftamologia gastou cerca de 100€.
% O José não sabe o valor certo do custo associado à consulta do amigo, mas sabe todas as
% informações restantes.

excecao(ato(data(30,04,2017),8,6,Custo,6)) :- cercade(Custo,100).
cercade(X,Y) :- A is 0.9*Y, B is 1.1*Y, X>=A, X=<B.
```

4.3.3 Conhecimento Interdito

Nesta situação o valor nulo para além de identificar um valor desconhecido, pretende significar que o valor que representa não é permitido especificar ou conhecer, qualquer

tentativa para concretizar este deverá ser rejeitada como sendo provocador de incosistência na informação presente na base de conhecimento.

```
% Impossibilidade de se saber a instituição de um determinado serviço.

nulo(instituicao_interdita).

servico(13,'Pediatria',instituicao_interdita,'Lisboa').
+servico(ID_Servico, Descricao, Instituicao, Cidade) :: (solucoes( (ID_Servico, Descricao, Instituicao_Interdita, Cidade),
                                                         (servico(13,'Pediatria',Instituicao_Interdita,'Lisboa'),nao(nulo(Instituicao_Interdita))), S),
                                                         comprimento(S,N), N == 0 ).

servico(14,'Gastrenterologia',instituicao_interdita,'Porto').
+servico(ID_Servico, Descricao, Instituicao, Cidade) :: (solucoes( (ID_Servico, Descricao, Instituicao_Interdita, Cidade),
                                                         (servico(14,'Gastrenterologia',Instituicao_Interdita,'Porto'),nao(nulo(Instituicao_Interdita))), S),
                                                         comprimento(S,N), N == 0 ).


excecao(servico(ID_Servico, Descricao, Instituicao, Cidade)) :- servico(ID_Servico, Descricao, instituicao_interdita, Cidade).
```

4.4 Manipulação de Invariantes que designem restrições à inserção e remoção de conhecimento

Para que a nossa base de conhecimento funcione corretamente é necessário implementar uma série invariantes, os quais são responsáveis por controlar a inserção e remoção de conhecimento sobre a mesma. Grande parte destes invariantes foram já desenvolvidos na primeira parte do trabalho prática, portanto, faremos referência aos invariantes que considerarmos relevantes para a realização deste.

- **Inserção de conhecimento - Invariantes de inserção**

Existe também um invariante de inserção para a data, e o que este faz é verificar se esta se encontra correta para ser inserida no sistema, ou seja, se o dia está compreendido entre 1 e 31, o mês entre 1 e 12 e o ano superior a 0.

```
% dias válidos para a data
+data(Dia, Mes, Ano) :: (Dia>=0; Dia=<31; Mes>=0; Mes=<12; Ano>=0).
```

Por ultimo, temos os invariantes que controlam a inserção de conhecimento interdito, ou seja, para um determinado parâmetro ou conjunto de parâmetros que consideremos ser de cariz interdito, o que este faz é não permitir que este seja inserido, certificando-se assim da nulidade do mesmo.

```
+servico(ID_Servico, Descricao, Instituicao, Cidade) :: (solucoes( (ID_Servico, Descricao, Instituicao_Interdita, Cidade),
                                                         (servico(13,'Pediatria',Instituicao_Interdita,'Lisboa'),nao(nulo(Instituicao_Interdita))), S),
                                                         comprimento(S,N), N == 0 ).
```

Neste exemplo, o invariante irá garantir que nunca seja possível ter conhecimento da instituição associada ao serviço com ID 13, de Pediatria, na cidade de Lisboa.

```
% Especializações invalidas/incompletas

nulo('Cirurgia').
nulo('Medicina').
nulo('Bombeiro').

+medico(ID_Medico, Nome, Idade, Morada, Especializacao) :: (ID_Medico \= medico_desconhecido, nao(nulo(Especializacao))).
```

Neste caso, um pouco mais restringente, o invariante irá garantir que nunca seja possível inserir um médico com ID desconhecido e com especialização em Cirurgia, Medicina ou Bombeiro, devido à incompletude e impertinência das mesmas.

- **Remoção de conhecimento - Invariantes de remoção**

No momento da remoção é necessário controlar a existência do conhecimento a ser removido (utente, servico, medico e ato):

```
-utente(IDU,N,I,M) :: (solucoes(IDU,utente(IDU,_,_,_),S),
                       comprimento(S,X),
                       X == 1).

-servico(IDS,D,I,C) :: (solucoes(IDS,servico(IDS,_,_,_),S),
                        comprimento(S,X),
                        X == 1).

-medico(ID,N,I,M,E) :: (solucoes(ID,medico(ID,_,_,_,_),S),
                        comprimento(S,X),
                        X == 1).

-ato(D,IDU,IDS,C,IDMED) :: (solucoes((D,IDU,IDS),ato(D,IDU,IDS,_,IDMED),S),
                            comprimento(S,X),
                            X==1).
```

Para além disso é necessário ter em atenção a consistência da base de conhecimento, isto é, não permitir a remoção de conhecimento que possui referência noutro lado. Estes casos acontecem com o ato, que possui os ID's do utente, servico e do médico. Para tal, é necessário verificar antes da remoção, se existe algum ato com o ID de um predicado, caso exista, não deve ser removido.

```
-utente(IDU,N,I,M) :: (solucoes(IDU,ato(_,IDU,_,_,_),S),
                       comprimento(S,X),
                       X==0).

-servico(IDS,D,I,C) :: (solucoes(IDS,ato(_,_,IDS,_,_),S),
                        comprimento(S,X),
                        X==0).

-medico(ID,N,I,M,E) :: (solucoes(ID,ato(_,_,_,_,ID),S),
                        comprimento(S,X),
                        X==0).
```

4.5 Problemática da evolução do conhecimento

O tratamento da problemática da evolução do conhecimento prende-se com o facto de deixar a base de conhecimento coesa e fidedigna a cada inserção ou remoção de conhecimento nesta.

Para garantir que isto aconteça não podemos permitir apagar informação que seja dependente de outra (p.e.: serviço incluído num ou mais atos médicos), ou permitir a inserção de informação repetida. Assim, aquando alteração da informação presente na base de conhecimento teremos de testar se a alteração pretendida não corremperá o conhecimento desta. Isto é conseguido com a utilização dos invariantes para inserção e remoção anteriormente abordados, tantos nos predicados de inserção como nos predicados de remoção.

```
remove(T) :- retract(T).

insercao(T) :- assert(T).
insercao(T) :- retract(T), !,fail.

testar([]).
testar([I|L]) :- I, testar(L).

evolucao(F) :-
    solucoes(I,+F::I,L),
    insercao(F),
    testar(L).

retrocesso(F) :-
    solucoes(I,-F::I,L),
    testar(L),
    remove(F).

solucoes(X,Y,Z) :- findall(X,Y,Z).
```

4.6 Desenvolvimento de um sistema de inferência capaz de implementar mecenismos de raciocínio inerentes a estes sistemas

O desenvolvimento de um sistema de inferência capaz de implementar os mecanismos de raciocínio inerentes aos sistemas anteriormente representados é muito importante tendo em conta que é este que pode efetivamente funcionar como um interpretador de questões, que mediante uma questão apresentada nos apresentará a conclusão do domínio de clusões aceitáveis: **verdadeiro**, **falso** ou **desconhecido**.

O conceito utilizado neste sistema foi o do Princípio do Mundo Fechado, que tal como vimos anteriormente, exige que só a informação mencionada na base de conhecimento seja verdadeira, considerando toda a restante falsa.

O funcionamento do interpretador presente a seguir é relativamente simples: dada uma questão este determinára a sua veracidade conforme o conhecimento presente na sua base de conhecimento. Assim, a resposta a uma questão será **verdadeiro** caso o conhecimento desta esteja presente; será **falso** caso esteja presente explicitamente a sua negação; ou, por fim, será **desconhecido** caso não esteja presente informação na base de conhecimento tornando esta verdadeira e não tenha sido explicitamente negada (*negação forte*).

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

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

5. *Conclusões*

Sendo este o segundo exercício do trabalho prático da Unidade Curricular, foi, de certa forma, mais acessível para nós relativamente ao primeiro, uma vez que já havíamos realizado o projeto anterior na linguagem de programação lógica *PROLOG*, e desta forma, já tínhamos algum do conhecimento necessário para iniciarmos a resolução do presente exercício.

Apesar de estarmos mais ambientados àquilo que são os sistemas de representação de conhecimento, foram, inevitavelmente, surgindo algumas dificuldades no que toca à descoberta de estratégias que melhor se ajustassem à implementação de alguns requisitos. A nossa maior dificuldade foi conseguir implementar de forma mais correta possível os três tipos de conhecimento imperfeito e ao mesmo tempo implementar os invariantes que com estes devem surgir. No entanto, depois de alguma dedicação e esforço, conseguimos ultrapassar esta dificuldade e várias outras com que mais à frente nos deparamos.

Assim, a realização deste exercício foi bastante importante para continuarmos a adquirir experiência e conhecimento, à medida que vamos entendendo como é que esta linguagem de programação lógica funciona e que problemas podem ser resolvidos e analisados com ela. Conseguimos, desta forma, consolidar não só todos os conhecimentos adquiridos nas aulas práticas, como também vários outros que foram sendo necessários durante a realização deste exercício.

Concluindo, no geral estamos satisfeitos e orgulhosos com o trabalho aqui desenvolvido e esperemos que assim continue nos próximos que se avizinham.

6. *Referências*

[Analide, 2011] ANALIDE, César, NOVAIS, Paulo, NEVES, José, "Sugestões para a Redação de Relatórios Técnicos", Departamento de Informática, Universidade do Minho, Portugal, 2011.

[Carlsson, 2011] CARLSSON, Mats, "SICStus Prolog User's Manual", Swedish Institute of Computer Science, Kista, Suécia, 2011.

[Analide, 1996] ANALIDE, César, NEVES, José, "Representação de Informação Incompleta", Departamento de Informática, Universidade do Minho, Portugal, 2011.