Sistemas de Representação de Conhecimento e Raciocínio
Exercício 1
Relatorio de Desenvolvimento

Grupo 18

Cesário Miguel Pereira Perneta - A73883
Luís Miguel Bravo Ferraz - A70824
Tiago Miguel Fraga Santos - A74092
Rui Pedro Barbosa Rodrigues - A74572

16 de Março de 2018

Conteúdo

Capítulo 1

Introdução

Neste primeiro exercício do trabalho de grupo, temos como objectivo desenvolver um sistema de representação de conhecimento e raciocínio que tenha a capacidade de caracterizar um universo de discurso na área da prestação de cuidados de saúde. Para tal usou-se o método sugerido no enunciado para a caracterização de conhecimento, onde temos *utente*, *prestador* e *cuidado*, os quais foram usados para a demonstração das várias funcionalidades.

Capítulo 2

Desenvolvimento

Neste capítulo iremos explicar o trabalho que foi desenvolvido, através de exemplos de codigo seguido da explicação do mesmo.

2.1 Base de Conhecimento

2.1.1 Utentes

utente: #IdUt, Nome, Idade, Morada $\rightarrow$ {*V,F*}

- utente(1,'Tiago',23,morada('Rua 1','Esposende')).
- utente(2,'Cesario',21,morada('Rua 2','Valença')).
- utente(3,'Luis',22,morada('Rua 3','Viana')).
- utente(4,'Rui',21,morada('Rua 4','Braga')).
- utente(5,'Marega',27,morada('Rua 5','Porto')).
- utente(6,'Tiquinho',25,morada('Rua 6','Porto')).
- utente(7,'Brahimi',26,morada('Rua 7','Olival')).
- utente(8,'Aboubakar',26,morada('Rua 8','Olival')).
- utente(9,'Danilo',25,morada('Rua 9','Porto')).
- utente(10,'Herrera',25,morada('Rua 10','Porto')).

2.1.2 Prestador

prestador: #IdPrest, Nome, Especialidade, Instituição $\rightarrow$ {*V,F*}

- prestador(1,'Nelon Puga','Desporto',instituicao('Clinica do Dragao', morada('Rua das Antas','Porto'))).
- prestador(2,'Joaquim','Cardiologia',instituicao('Clinica do Dragao', morada('Rua das Antas','Porto'))).

- prestador(3,'Antonio','Fisioterapia',instituicao('Clinica do Dragao', morada('Rua das Antas','Porto'))).
- prestador(4,'Pedro','Radiologia',instituicao('Clinica do Dragao', morada('Rua das Antas','Porto'))).
- prestador(5,'Manuel','Oftalmogia',instituicao('Clinica do Dragao', morada('Rua das Antas','Porto'))).
- prestador(6,'Ines','Cardiologia',instituicao('Hospital Braga', morada('Rua do Hospital','Braga'))).
- prestador(7,'Joana','Cirurgia',instituicao('Hospital Braga', morada('Rua do Hospital','Braga'))).
- prestador(8,'Maria','Radiologia',instituicao('Hospital Braga', morada('Rua do Hospital','Braga'))).
- prestador(9,'Diana','Oftalmogia',instituicao('Hospital Braga', morada('Rua do Hospital','Braga'))).
- prestador(10,'Ana','Fisioterapia',instituicao('Hospital Braga', morada('Rua do Hospital','Braga'))).
- prestador(11,'Carlos','Cardiologia',instituicao('Hospital Sao Joao', morada('Rua da Circunvalacao','Porto'))).
- prestador(12,'Tomas','Cirurgia',instituicao('Hospital Sao Joao', morada('Rua da Circunvalacao','Porto'))).
- prestador(13,'Raquel','Radiologia',instituicao('Hospital Sao Joao', morada('Rua da Circunvalacao','Porto'))).
- prestador(14,'Luisa','Oftalmogia',instituicao('Hospital Sao Joao', morada('Rua da Circunvalacao','Porto'))).
- prestador(15,'Teresa','Fisioterapia',instituicao('Hospital Sao Joao', morada('Rua da Circunvalacao','Porto'))).

2.1.3 Utentes

cuidado: Data, #IdUt, #IdPrest, Descrição, Custo $\rightarrow$ { *V,F*}

- cuidado(data(10,02,2018),1,9,'Consulta Rotina',30).
- cuidado(data(10,02,2018),2,9,'Consulta Rotina',30).
- cuidado(data(10,02,2018),3,9,'Consulta Rotina',30).
- cuidado(data(10,02,2018),4,9,'Consulta Rotina',30).
- cuidado(data(11,02,2018),5,9,'Consulta Rotina',30).
- cuidado(data(15,02,2018),5,1,'Consulta Pos-Match',10).
- cuidado(data(15,02,2018),6,1,'Consulta Pos-Match',10).

- cuidado(data(15,02,2018),7,1,'Consulta Pos-Match',10).
- cuidado(data(15,02,2018),8,1,'Consulta Pos-Match',10).
- cuidado(data(15,02,2018),9,1,'Consulta Pos-Match',10).
- cuidado(data(16,02,2018),5,11,'Eletrocardiograma',20).
- cuidado(data(16,02,2018),6,11,'Eletrocardiograma',20).
- cuidado(data(16,02,2018),7,11,'Eletrocardiograma',20).
- cuidado(data(17,02,2018),10,15,'Fisio - Coxa Direita',10).
- cuidado(data(17,02,2018),8,15,'Fisio - Gemeo Esq',10).
- cuidado(data(18,03,2018),10,13,'Raio X',50).
- cuidado(data(10,03,2018),5,1,'Consulta Pos-Match',10).
- cuidado(data(10,03,2018),7,1,'Consulta Pos-Match',10).
- cuidado(data(10,03,2018),8,1,'Consulta Pos-Match',10).
- cuidado(data(10,03,2018),9,1,'Consulta Pos-Match',10).
- cuidado(data(12,03,2018),6,15,'Fisio - Coxa Esq',10).
- cuidado(data(12,03,2018),7,15,'Fisio - Joelho Dir',10).
- cuidado(data(12,03,2018),8,15,'Fisio - Costas',10).
- cuidado(data(12,03,2018),9,15,'Fisio - Virilha',10).
- cuidado(data(12,03,2018),10,15,'Fisio - Coxa Dir',10).
- cuidado(data(20,02,2018),1,9,'Consulta',30).
- cuidado(data(20,02,2018),2,6,'Eletrocardiograma',25).
- cuidado(data(20,02,2018),3,7,'Intervenção',30).
- cuidado(data(20,02,2018),4,8,'Consulta Rotina',30).
- cuidado(data(20,02,2018),5,5,'Consulta Rotina',30).

2.1.4 Instituição

instituicao: Nome,Morada -¿ V,F

- instituicao('Clinica do Dragao', morada('Rua das Antas','Porto')).
- instituicao('Hospital Sao Joao', morada('Rua da Circunvalacao','Porto')).
- instituicao('Hospital Braga', morada('Rua do Hospital','Braga')).
- instituicao('Centro Saude', morada('Rua Centro','Braga')).
- instituicao('Hospital Viana', morada('Rua do Hospital','Viana')).
- instituicao('Hospital Esposende', morada('Rua do Hospital', 'Esposende')).

2.2 Funcionalidades

2.2.1 Predicados Auxiliares

Estes predicados foram desenvolvidos para auxiliar na resolução dos predicados principais. Alguns deles fomos nós que desenvolvemos sendo que outros foram desenvolvidos nas aulas teóricas e nas teorico práticas.

```
data(D, M, A) :- pertence(M, [1,3,5,7,8,10,12]), D >= 1, D =< 31.
data(D, M, A) :- pertence(M, [4,6,9,11]), D >= 1, D =< 30.
data(D, 2, A) :- A mod 4 =\= 0, D >= 1, D =< 28.
data(D, 2, A) :- A mod 4 =:= 0, D >= 1, D =< 29.
```

Este predicado é usado para validar datas, isto é, o dia (D), o mês (M) e o ano (A).

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

Este predicado insere o conhecimento na base de conhecimento em caso de sucesso, no entanto, em caso de insucesso remove o conhecimento inserido.

```
remocao(T) :- retract(T).
remocao(T) :- assert(T),!,fail.
```

Este predicado remove o conhecimento da nossa base de conhecimento.

```
teste([]).
teste([I|L]) :- I, teste(L).
```

Este predicado testa se existe algum invariante que não seja satisfeito. Se falhar algum invariante, então a base de conhecimento não vai evoluir, caso todos forem satisfeitos, a base de conhecimento vai evoluir.

```
comprimento([],0).
comprimento([X|L],R) :- comprimento(L,N) , R is 1+N.
```

Este predicado calcula o comprimento de uma lista.

```
unicos([],[]).
unicos([H|T], R) :-
    pertence(H,T),
    unicos(T,R).
unicos([H|T], [H|R]) :-
    nao(pertence(H,T)),
    unicos(T,R).
```

Este predicado verifica se os elementos que compõem uma lista são todos diferentes.

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

Este predicado verifica se um conhecimento existe na nossa base de conhecimento.

```
pertence(H,[H|T]).
pertence(X,[H|T]) :-
    X \= H,
    pertence(X,T).
```

Este predicado verifica se elemento pertence a uma lista.

```
somaTotal([],0).
somaTotal([X],X).
somaTotal([X|L],R) :- somaTotal( L,R1 ), R is X+R1.
```

Este predicado soma todos os elementos de uma lista.

```
solucoes(T,Q,S) :- findall(T,Q,S).
```

Este predicado utiliza o predicado findall disponibilizado pelo PROLOG, de modo a colocar numa lista as solucões a questão efetuada.

2.2.2 Registar utentes, prestadores e cuidados de saúde

Neste ponto pretende-se registar utentes, prestadores e cuidados de saúde, tendo em conta certos casos onde a inserção será inválida.

Invariantes

```
+utente(ID,N,I,M) ::
    (solucoes( (ID), (utente(ID,Ns,Is,Ms)),S),
    comprimento(S,N),
    N==1).
```

Invariante Estrutural: não permitir a inserção de conhecimento repetido de utentes. Pode estar repetido de duas formas:

⇒ ID

⇒ Nome

```
+prestador(ID,N,E,I) ::
    (solucoes( (ID), (prestador(ID,Ns,Es,Is)), S),
    comprimento(S,N),
    N==1).
```

Invariante Estrutural: não permitir a inserção de conhecimento repetido de prestadores. Pode estar repetido de duas formas:

⇒ ID

⇒ Nome

```
+cuidado(Dt,IDU,IDP,D,C) ::
    (solucoes( (Dt,IDU,IDP),(cuidado(Dt,IDU,IDP,Ds,Cs)),S),
    comprimento(S,N),
    N==1).
```

Invariante Estrutural: não permitir a inserção de conhecimento repetido de cuidados. Pode estar repetido, quando se verifica em simultâneo igualdades na:

⇒ Data

⇒ Id utente

⇒ Id prestador

```
+cuidado(Dt,IDU,IDP,D,C) ::
    (solucoes( (IDU),(utente(IDU,Ns,Is,Ms)),S),
    comprimento(S,N),
    N==0).
+cuidado(Dt,IDU,IDP,D,C) ::
    (solucoes( (IDP),(prestador(IDP,Ns,Es,Is)),S),
    comprimento(S,N),
    N==0).
```

Invariante Referencial: não permitir a inserção de conhecimento, por falta de conhecimento de cuidados. Não se insere o conhecimento quando não há conhecimento de:

⇒ Utente

⇒ Prestador

```
+instituicao(N,M) ::
    (solucoes( (N),instituicao(N,M) ,S),
    N==1).
```

Invariante Estrutural: não permitir a inserção de conhecimento repetido de instituições. Pode estar repetido de duas formas:

⇒ Pelo Nome da instituição.

```
-instituicao(N,M) ::
    (solucoes( (N,M),prestador(\_,\_,\_,(N,M)) , S),
    N==0).
```

Invariante Estrutural: não permitir a remoção de conhecimento. Não se pode remover conhecimento de prestador quando:

⇒ Há conhecimento de instituicao nos prestadores.

Predicado Principal

```
registar( Termo ) :-
    solucoes( Invariante,+Termo::Invariante,Lista),
    insercao(Termo),
    teste(Lista).
```

Este predicado é responsável por registar utentes, prestadores e cuidados de saúde, tendo em consideração os invariantes que já foram mensionados e usando os predicados auxiliares. Este predicado é igual ao predicado evolução que foi desenvolvido na aula teórica.

2.2.3 Remover utentes, prestadores e cuidados de saúde

Neste ponto pretende-se remover utentes, prestadores e cuidados de saúde, tendo em conta certos casos onde a remoção será inválida.

Invariantes

```
-utente(ID,N,I,M) ::
    (solucoes( (ID), (cuidado(Dts,ID,IDPs,Ds,Cs)) , S),
    comprimento(S,N),
    N>0).
```

Invariante Estrutural: não permitir a remoção de conhecimento de utentes. Não se pode remover conhecimento de utente quando:

⇒ Há conhecimento de utente nos cuidados.

```
-prestador(ID,N,E,I) ::
    (solucoes( (ID), (cuidado(Dts,IDUs,ID,Ds,Cs)), S),
    comprimento(S,N),
    N==0).
```

Invariante Estrutural: não permitir a remoção de conhecimento de prestadores. Não se pode remover conhecimento de prestador quando:

⇒ Há conhecimento de prestador nos cuidados.

```
-cuidado(Dt,IDU,IDP,D,C) ::
    (solucoes( (Dt,IDU,IDP,D,C), (cuidado(Dt,IDU,IDP,D,C)) ,S),
    comprimento(S,N),
    N==0).
```

Invariante Estrutural: não permitir a remoção de conhecimento de cuidados. Não se pode remover conhecimento de cuidado quando:

⇒ Não há esse conhecimento de cuidado

Predicado Principal

```
remover( Termo ) :-
    solucoes( Invariante,+Termo::Invariante,Lista),
    remocao(Termo),
    teste(Lista).
```

Este predicado é responsável por remover utentes, prestadores e cuidados de saúde, tendo em consideração os invariantes que já foram mencionados e usando predicados auxiliares. Este predicado é igual ao predicado involução que foi referenciado na aula teórica.

2.2.4 Identificar utentes por critérios de seleção

Predicado Principal

Este predicado vai identificar utentes através de vários critérios de seleção como o seu ID, nome, idade ou morada.

```
ponto_tres(ids,R) :- solucoes((ID),utente(ID,Ns,Is,Ms),R).
ponto_tres(nomes,R) :- solucoes((N),utente(IDs,N,Is,Ms),R).
ponto_tres(idades,R) :- solucoes((I),utente(IDs,Ns,I,Ms),R).
ponto_tres(moradas,R) :- solucoes((M),utente(IDs,Ns,Is,M),R).
```

Conclusão do predicado

1. Lista com ids dos utentes;
2. Lista com nomes dos utentes;
3. Lista com idades dos utentes;
4. Lista com maradas dos utentes

Extensão do predicado

No predicado solucoes, no segundo argumento, queremos encontrar todas as ocorrencias que tenham um argumento igual ao do primeiro argumento, neste caso o (ID)/(N)/(I)/(M) e colocamos todas as ocorrencias que correspondem a esta comparação no terceiro argumento do predicado soluçoes.

2.2.5 Identificar as instituições prestadoras de cuidados de saúde

Predicado Principal

Este predicado identica as instituicões que prestam cuidados de saúde.

```
ponto_quatro(todas,R) :-
    solucoes((N), instituicao(N,M), R).
ponto_quatro(cuidados,R) :-
    solucoes((N), (cuidado(_,_,IDP,_,_),prestador(IDP,_,_,instituicao(N,M))) , S),
    unicos(S,R).
```

Conclusão do predicado

1. Lista com o NOME de todas as instituiçoes que estao na nossa base de conhecimento, pq na nossa base de conhecimento o que define uma instituição achamos que devia ser o nome.

2. Lista com o NOME de todas as instuiçoes que estao referenciadas em predicados de cuidados, pq na nossa base de conhecimento o que define uma instituição achamos que devia ser o nome.

Extensão do predicado

1. No predicado solucoes, no segundo argumento, queremos encontrar todas as ocorrencias que tenham um argumento igual ao do primeiro argumento, neste caso o (N) e colocamos todas as ocorrencias que correspondem a esta comparação no terceiro argumento do predicado soluçoes.

2. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de cuidados e vamos so guardar o seu 'IDP', e fazemos uma comparação com todas as ocorrências de prestadores cujo o IDP seja o mesmo. Por fim guardamos o Nome(N) da instituição (pois, foi o que dissemos que queriamos guardar, a partir do primeiro argumento) numa lista auxiliar. Em seguida passamos essa lista pelo predicado 'unicos' para 'limpar' o resultado e só mostrar uma vez o nome de cada instituição.

2.2.6 Identificar cuidados de saúde prestados por instituição/cidade/datas

Predicado Principal

Este predicado identifica os cuidados prestados por uma dada instituição,cidade ou data.

```
ponto_cinco(inst,X,R) :-
    solucoes((Dt,IDU,IDP,D,C), (prestador(IDP,_,_,instituicao(X,_)), cuidado(Dt,IDU,IDP,D,
    unicos(S,R).
ponto_cinco(cidade,X,R) :-
    solucoes( (Dt,IDU,IDP,D,C) , (prestador(IDP,_,_,instituicao(_,morada(_,X))) , cuidado(
    unicos(S,R).
ponto_cinco(data,X,R) :- solucoes( (X,IDU,IDP,D,C) , cuidado(X,IDU,IDP,D,C) , R).
```

Conclusão do predicado

1. Lista com todos os cuidados de saúde que tenham sido feitos na instituição que é fornecida á conclusão.

2. Lista com todos os cuidados de saúde que tenham sido feitos na cidade que é fornecida á conclusão.

3. Lista com todos os cuidados de saúde que tenham sido feitos na data que é fornecida á conclusão.

Decidimos que a resposta apresentada, ou seja, a lista com todos os cuidados, estes iam ser definidos pela data, o IDU, o IDP, a descrição e o custo, ou seja todos os seus argumentos, definimos portanto que para definir um cuidado de saúde, este so é definido por todos os seus argumentos.

Extensão do predicado

1. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de prestadores cujo nome da instituição seja igual ao colocado na conclusão, 'guardamos' o IDP de todas as ocorrencias que obedeçam a este critério para fazer um match com todos os cuidados que contenham um IDP igual ao referido. Por fim, como no primeiro argumento definimos que queriamos todos os argumentos dos cuidados, estes sao guardados no resultado

2. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de prestadores cuja cidade da morada da instituição seja igual ao colocado na conclusão, 'guardamos' o IDP de todas as ocorrencias que obedeçam a este critério para fazer um match com todos os cuidados que contenham um IDP igual ao referido. Por fim, como no primeiro argumento definimos que queriamos todos os argumentos dos cuidados, estes sao guardados no resultado

3. No predicado solucoes, no segundo argumento, queremos encontrar todas as ocorrencias que tenham um argumento igual ao do primeiro argumento, neste caso a dat e colocamos todas as ocorrencias que correspondem a esta comparação no terceiro argumento do predicado soluçoes.

2.2.7 Identificar os utentes de um prestador/especialidade/instituição

Predicado Principal

Este predicado identifica os utentes dada um prestador, especialidade ou instituição, onde X que representa o que queremos procurar e R é onde é guardado o resultado final.

```
ponto_seis(prest,X,R) :-
    solucoes( (IDU,N), ( cuidado(Dt,IDU,X,D,C),utente(IDU,N,_,_)), S),
    unicos(S,R).
ponto_seis(esp,X,R) :-
    solucoes( (IDU,N), ( prestador(IDP,_,X,_), cuidado(_,IDU,IDP,_,_), utente(IDU,N,_,_)),
    unicos(S,R).
ponto_seis(inst,X,R) :-
    solucoes( (IDU,N), ( cuidado(_,IDU,_,_,instituicao(X,_)), utente(IDU,N,_,_)), S),
    unicos(S,R).
```

Conclusão do predicado

1. Lista com todos os utentes que tenham tido cuidados de saúde com o prestador cujo ID é fornecido na conclusão, pois na nossa base de conhecimento os ids sao unicos, logo nao existe dois prestadores com o mesmo id, enquanto que pode haver prestadores com o mesmo nome.

2. Lista com todos os utentes que tenham tido cuidados de saúde na especialidde que é fornecida á conclusão.

3. Lista com todos os utentes que tenham tido cuidados de saúde na instiuiçao cujo nome é fornecido à conclusão, pq como dissemos anteriormente o maior ponto que define uma instituição é o nome.

Decidimos que na resposta apresentada, ou seja, a lista com todos os utentes, os utentes iam ser definidos pelos dois pontos cruciais que os definiem, ou seja, cada utente é definido pelo seu id e pelo seu nome, visto que nao há ids repetidos na nossa base de conhecimento enquanto que nome pode haver, e para definir um utente so pelo id pareceu-nos pouca informação.

Extensão do predicado

1. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrências de cuidados cujo id do prestador seja igual ao colocado na conclusão, 'guardamos' o IDU de todas as ocorrências que obedeçam a este critério para fazer um match com todos os utentes da nossa base de conhecimento que contenham um IDU igual ao referido. Por fim, como no primeiro argumento definimos que queriamos o IDU e o Nome dos utentes, estes sao guardados numa lista auxiliar, para depois serem passados no predicado unicos

2. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de prestadores cuja especialidade seja igual ao colocado na conclusão, 'guardamos' o IDP de todas as ocorrencias que obedeçam a este critério para fazer um match com todos os cuidados que contenham um IDP igual ao referido. A seguir 'guardamos' o IDU de todos os cuidados que fizeram o match referido para voltar a fazer um match com as ocorrencias de todos os utentos. Por fim, como no primeiro argumento definimos que queriamos o IDU e o Nome dos utentes, estes sao guardados numa lista auxiliar, para depois serem passados no predicado únicos.

3. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de cuidados cujo nome da instituiçãoseja igual ao colocado na conclusão, 'guardamos' o IDU de todas as ocorrencias que obedeçam a este critério para fazer um match com todos os utentes que contenham um IDU igual ao referido. Por fim, como no primeiro argumento definimos que queriamos o IDU e o Nome dos utentes, estes sao guardados numa lista auxiliar, para depois serem passados no predicado únicos.

2.2.8 Identificar cuidados de saúde realizados por utente/instituição/prestador

Predicado Principal

Este predicado identica todos os cuidados realizados por um utente,instituicão ou prestador, onde X que representa o que queremos procurar e R é onde é guardado o resultado final.

```
ponto_sete(utente,X,R) :-
    solucoes( (Dt,X,IDP,D,C), cuidado(Dt,X,IDP,D,C), R).
ponto_sete(inst,X,R) :-
    solucoes( (Dt,IDU,IDP,D,C), ( prestador(IDP,_,_,instituicao(X,_)), cuidado(Dt,IDU,IDP,
ponto_sete(prest,X,R) :-
    solucoes( (Dt,IDU,X,D,C), cuidado(Dt,IDU,X,D,C), R).
```

Conclusão do predicado

1. Lista com todos os cuidados de saúde que tenham sido feitos ao utente que é fornecida á conclusão. Na conclusão é fornecido o ID pq é o identificador único dos utentes na nossa base de conhecimento.

2. Lista com todos os cuidados de saúde que tenham sido feitos na instituição que é fornecida á conclusão.Na conclusão é fornecido o nome pq é o identificador único das instituições na nossa base de conhecimento.

3. Lista com todos os cuidados de saúde que tenham sido feitos pelo prestador que é fornecida á conclusão. Na conclusão é fornecido o ID pq é o identificador único dos prestadores na nossa base de conhecimento.

Decidimos que a resposta apresentada, ou seja, a lista com todos os cuidados, estes iam ser definidos pela data, o IDU, o IDP, a descrição e o custo, ou seja todos os seus argumentos, definimos portanto que para definir um cuidado de saúde, este so é definido por todos os seus argumentos.

Extensão do predicado

1. No predicado solucoes, no segundo argumento, vamos buscar todas as ocorrencias cujo IDU seja igual ao que esta na conclusão. Como no primeiro argumento do predicado solucoes estao todos os paramentos do predicado cuidado, pois é assim que definimos os cuidados, o resultado vai ser essa lista.

2. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de prestador cujo nome da instituição seja igual ao colocado na conclusão, 'guardamos' o IDP de todas as ocorrencias que obedeçam a este critério para fazer um match com todos os cuidados que contenham um IDP igual ao referido. Por fim, como no primeiro argumento definimos que queriamos todos os argumentos do predicado cuidado e estes sao guardados na resposta.

3. No predicado solucoes, no segundo argumento, vamos buscar todas as ocorrencias cujo IDP seja igual ao que esta na conclusao. Como no primeiro argumento do predicado solucoes estao todos os paramentos do predicado cuidado, pois é assim que definimos os cuidados, o resultado vai ser essa lista.

2.2.9 Determinar todas as instituições/prestadores a que um utente já recorreu

Predicado Principal

Este predicado indica todas as instituices e servicos a que um utente ja recorreu, onde U que representa o que queremos procurar e R é onde é guardado o resultado final.

```
ponto_oito(inst,U,R) :-
    solucoes( (N) , ( cuidado(_,U,IDP,_,_), prestador(IDP,_,_,instituicao(N,_))) ,S),
    unicos(S,R).
ponto_oito(inst,U,R) :-
    solucoes( (ID,N) , ( cuidado(_,U,IDP,_,_), prestador(IDP,N,_,_) ,S)),
    unicos(S,R).
```

Conclusão do predicado

1. Lista com todos as instituições que tenham sido frequentadas por um utente. Na conclusao é inserido o ID do utente pois é o identificador unico do utente na nossa base de conhecimento. A resposta é uma lista com o nome das instituições pois, é o identificador unico das instituiçoes da nossa base de conhecimento.

2. Lista com todos os prestadores de saúde que tenham prestado serviços a um dado utente. Na conclusão é inserido o ID do utente pois é o identificador único do utente na nossa base de conhecimento. A resposta é uma lista com o id e o nome dos prestadores, o id é o identificador único dos prestadores na nossa base de conhecimento, no entanto, achamos que era curto para definir com informação suficiente um prestador, como tal, decidimos descreve-lo com o id e o nome, este último nao é um identificador unico.

Extensão do predicado

1. No predicado solucões, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de cuidado cujo IDU seja igual ao colocado na conclusão, 'guardamos' o IDP de todas as ocorrencias que obedeçam a este critério para fazer um match com todos os prestadores que contenham um IDP igual ao referido, e como no primeiro argumento do predicado solucçoes especificamos que queremos o nome das instituiçoes, colocamos em evidencia no segundo argumento o que pretendemos para ser guardado na lista auxiliar para ser passada a limpo no predicado unicos.

2. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de cuidado cujo IDU seja igual ao colocado na conclusão, 'guardamos' o IDP de todas as ocorrencias que obedeçam a este critério para fazer um match com todos os prestadores que contenham um IDP igual ao referido, e como no primeiro argumento do predicado solucçoes especificamos que queremos o id e o nome do prestador, colocamos em evidencia no segundo argumento o que pretendemos para ser guardado na lista auxiliar para ser passada a limpo no predicado unicos.

2.2.10 Calcular o custo total dos cuidados de saúde por utente/especialidade/prestador/datas

Predicado Principal

Este predicado calcula o custo total dos cuidados de saúde dado um utente, especialidade, prestador ou datas, onde X que representa o que queremos procurar e R é onde é guardado o resultado final.

```
ponto_nove(utente,X,R) :-
    solucoes( (C), cuidado(Dt,X,IDP,D,C), S),
    somaTotal(S,R).
ponto_nove(esp,X,R) :-
    solucoes( (C), (prestador(IDP,_,X,_), cuidado(_,_,IDP,_,C)) , S),
    somaTotal(S,R).
ponto_nove(prest,X,R) :-
    solucoes( (C), cuidado(Dt,IDU,X,D,C), S),
    somaTotal(S,R).
ponto_nove(data,X,R) :-
    solucoes( (C), cuidado(X,IDU,IDP,D,C), S),
    somaTotal(S,R).
```

Conclusão do predicado

1. Valor da soma total de todos os cuidados de saúde efetuados por utente, passado como argumento na conclusão. Como o identificador unico dos utentes na nossa base de conhecimento é o ID, entao decidimos fornecer o ID á conclusao.

2. Valor da soma total de todos os cuidados de saúde efetuados por utente, passado como argumento na conclusão. A especialidade é um argumento do predicado prestador, logo tem de ser passada como esta na base de conhecimento.

3. Valor da soma total de todos os cuidados de saúde efetuados por um prestador, passado como argumento na conclusão. Como o identificador único dos prestadores na nossa base de conhecimento é o ID, entao decidimos fornecer o ID á conclusão.

4. Valor da soma total de todos os cuidados de saúde efetuados por utente, passado como argumento na conclusão. Aqui usa-se o predicado auxiliar data que já foi referido.

Extensão do predicado

1. No predicado solucoes, no segundo argumento, vamos buscar todas as ocorrencias cujo IDU seja igual ao que esta na conclusao. Como no primeiro argumento do predicado solucoes esta o argumento custo, guardamos todos os custos numa lista e fazemos a soma total da lista e colocamos o resultado no argumento R.

2. No predicado solucoes, no segundo argumento, vamos em primeiro lugar buscar TODAS as ocorrencias de prestador cujo IDP seja igual ao colocado

na conclusão, 'guardamos' o IDP de todas as ocorrencias que obedeçam a este critério para fazer um match com todos os cuidados que contenham um IDP igual ao referido, e como no primeiro argumento do predicado solucçoes especificamos que queremos custo dos cuidados, guardamos todos os custos numa lista e fazemos a soma total da lista e colocamos o resultado no argumento R.

3. No predicado solucoes, no segundo argumento, vamos buscar todas as ocorrencias cujo IDP seja igual ao que esta na conclusao. Como no primeiro argumento do predicado solucoes esta o argumento custo, guardamos todos os custos numa lista e fazemos a soma total da lista e colocamos o resultado no argumento R.

4. No predicado solucoes, no segundo argumento, vamos buscar todas as ocorrencias cuja data seja igual ao que esta na conclusao. Como no primeiro argumento do predicado solucoes esta o argumento custo, guardamos todos os custos numa lista e fazemos a soma total da lista e colocamos o resultado no argumento R.

Capítulo 3

Conclusão

Este primeiro exercício ajudou bastante a ambientarmo-nos à linguagem de programacão lógica PROLOG, permitindo-nos caracterizar um universo de forma simples e estruturada.

Os principais obstáculos que encontramos no desenvonvimento do projeto, foi na forma de como íriamos estruturar o nosso pensamento usando programacão em lógica, sendo que após ultrapassarmos este problema, a resolução do exercício foi fácil.

Em suma, cumprimos todos os requisitos pedidos no enunciado, embora poderíamos ter feito mais algumas funcionalidades extras para o trabalho ficar mais completo. Contudo trata-se de um trabalho competente.