

Universidade do Minho

Exercício 2

Licenciatura em Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio

2ºSemestre (2014/2015)

67673 André Gerales

67665 Patrícia Barros

67709 Sandra Ferreira

Braga

Março de 2015

Resumo

Este trabalho foi realizado no âmbito da Unidade Curricular de Sistemas de Representação de Conhecimento e Raciocínio e consiste no desenvolvimento de um sistema de representação de conhecimento imperfeito que caracterize um universo de comércio automóvel.

Índice

Introdução	5
Preliminares	6
Descrição do Trabalho	7
Caso Prático de Aplicação	7
Predicados	10
Valores Nulos	11
Inserção e Remoção de Conhecimento	12
Invariantes	13
Interação Com o Sistema em Java	14
Figura 1 - Menu da nossa aplicação	Erro! Marcador não definido.
Análise de Resultados	15
Conclusões e Sugestões	17

Índice de Figuras

Figura 1 Menu da aplicação	14
Figura 2 Resultado de pedir ao sistema o boletim de um automóvel	14
Figura 3 Resultado de pedir ao sistema todos os automóveis de cor azul.....	15
Figura 4 Resultado de pedir ao sistema os proprietários de uma automóvel até um determinado ano	15
Figura 5 Resultado da primeira query	15
Figura 6 Resultado da segunda query	16
Figura 7 Resultado da terceira query	16
Figura 8 Resultado da quarta query	16
Figura 9 Resultado da quinta query	16

Introdução

O trabalho prático descrito neste relatório consiste no desenvolvimento de um sistema de representação de conhecimento imperfeito que seja capaz de descrever um universo de comércio automóvel, recorrendo à utilização de valores nulos.

A linguagem utilizada para desenvolver este trabalho será a linguagem de programação lógica **PROLOG**.

Neste relatório apresentam-se o processo de desenvolvimento do sistema de representação de conhecimento e os resultados obtidos.

Preliminares

De forma a conseguirmos realizar o trabalho proposto foi necessário, através das aulas da Unidade Curricular de Sistemas de Representação de Conhecimento e Raciocínio, possuímos conhecimentos base de **PROLOG** e construção de mecanismos de raciocínio para resolução de problemas.

Descrição do Trabalho

Como referido anteriormente este trabalho consiste na realização de um sistema de representação de conhecimento e raciocínio que possibilite a descrição de um universo de comércio automóvel.

Caso Prático de Aplicação

Para que fosse possível demonstrar as capacidades do sistema desenvolvido foi necessário criar um caso prático de aplicação do cenário criado.

De forma a dar resposta ao enunciado criado e a tornar compreensivo e claro o trabalho desenvolvido, foi necessário inserir informação no ficheiro .pl.

Para tornar este enunciado mais interessante decidiu-se manipular um volume considerável de informação, tendo então como base o conhecimento de 8 automóveis.

Segue-se então a conhecimento inserido ficheiro.

O Xico é o construtor dos automóveis 1, 4 e 7. O Neves é também o construtor de 3 automóveis: os automóveis 2, 6 e 8. Os carros cujo id são 3 e 5 foram construídos pelo Rego.

Neste comércio existem automóveis da marca Toyota (automóvel 2, 4 e 5) e outros da marca Opel (3 e 6). O veículo 7 é da marca Mercedes e o 8 é da Nissan. Ainda sobre as marcas, outro tópico que é necessário abordar são os modelos relativos a cada marca. Optou-se por se trabalhar apenas com 3 tipos: o modelo PequenoCarro, o GrandeCarro e ainda o modelo Fininho. Posto isto, os carros 2 e 6 pertencem ao primeiro modelo referido, os carros 1, 4 e 5 são do segundo modelo e os restantes são do modelo Fininho.

Os automóveis são também classificados pelo estado em que se encontram: novo, semi-novo ou usado. Os carros cujo id são 1 e 5 são novos

enquanto os carros semi-novos são o 4 e o 8. Os restantes automóveis são usados.

Neste comércio existem carros das mais variadas cores: o carro 1 é verde, o 3, o 5 e o 7 são azuis, o carro 4 é rosa e o 6 e o 8 são pretos.

Outro conhecimento que foi inserido neste ficheiro é relativo ao ano de fabrico de cada um destes veículos. O veículo 4 é o mais antigo tendo sido fabricado em 2002. Os veículos 5 e 6 foram fabricados um ano após o veículo 4. Os automóveis 7, 8 e 1 são os mais recentes pertencendo ao ano de fabrico 2010, 2011 e 2015, respetivamente.

Por fim, o último tópico abordado foi os proprietários de cada automóvel. O automóvel com id=1 pertenceu unicamente ao Manel, sendo este proprietário desde 2015. Já o automóvel 2 teve dois proprietários o Tiago em 2010 e o jorge em 2013. O automóvel com id=3 conta agora com o terceiro proprietário, o Ricardo, desde 2013, tendo-lhe antecedido o Tozé em 2005 e o Ricardo em 2013.

A Rita é, desde 2008, proprietária do carro 5. Anteriormente este veículo tinha pertencido ao Manuel no ano de 2006. O carro 6 só conta com a Patrícia como proprietária, desde 2012. O carro 7 é propriedade do Joao desde 2011 e esteve na posse da Joana no ano anterior. Tal como o veículo 4, o veículo 8 já teve 3 proprietários: o André em 2008, a Sandra em 2015 e a Daniela em 2013.

Tem-se então apresentada a informação disponível para responder ao enunciado proposto pelo grupo.

Segue-se então o enunciado que criamos e seguimos para a realização dos projeto.

1. O Xico é o construtor do automóvel com identificador 1. Sabe-se que o modelo do carro é grandeCarro, mas desconhece-se a sua marca.
2. O automóvel, cujo id é 2, é da marca Toyota do modelo pequenoCarro. O construtor deste carro é o Neves.
3. O Rego construiu o automóvel com id=3, da marca Opel e modelo Fininho.

4. O automóvel 1 é verde e tem como matrícula xxaa11.
5. A matrícula do automóvel 2 é xxyy22, mas desconhece-se qual seja a sua cor.
6. O carro 3 é azul. Mas não se sabe se a matrícula do automóvel é xxzz33 ou xxzy33.
7. O automóvel do Xico (automóvel 1) está em bom estado. Diz-se portanto que o seu estado é novo.
8. O Neves não sabe se o estado do carro que construiu é novo ou semi-novo.
9. O carro construído pelo Rego é usado.
10. O Xico sabe que o ano de fabrico do carro que construiu é 2015.
11. O ano de fabrico do automóvel 2 não se sabe ao certo mas terá sido entre 1998 e 2000.
12. Nunca será possível conhecer o ano de fabrico do automóvel 3.
13. O proprietário do automóvel 1 é o Manel, tendo-o adquirido em 2015.
14. O Jorge comprou em 2013 o automóvel 2, tendo este pertencido anteriormente ao Tiago que o havia comprado em 2010.
15. A Sara possui agora o automóvel 3. Comprou-o em 2012.
16. Anteriormente à Sara, o automóvel 3 já tinha pertencido à Rita em 2010 e ao Zeca em 2005.

Predicados

Para o desenvolvimento do nosso sistema predicados relativos ao construtor de um automóvel, à sua marca, ao seu modelo, à sua matrícula, à sua cor, ao seu estado, ao ano em que foi fabricado e aos registos de venda que existem para ele. Para além destes criamos predicados que nos permitem saber, por exemplo, se uma certa pessoa foi proprietária de um certo carro desde ou até um determinado ano:

- predicado `automovelC:: Id, construtor -> {V,F,D}`
- predicado `automovelM:: Id, marca -> {V,F,D}`
- predicado `automovel:: Id, modelo -> {V,F,D}`
- predicado `matricula :: Id, matricula -> {V,F,D}`
- predicado `cor :: Id, cor -> {V,F,D}`
- predicado `cores: Id, Resultados -> {V,F}`
- predicado `estado :: Id, estado -> {V,F,D}`
- predicado `anofabrico :: Id, ano -> {V,F,D}`
- predicado `proprietario :: Id, nome, ano -> {V,F,D}`
- predicado `proprietarioDesde : Id, Nome, Ano -> {V,F,D}`
- predicado `proprietarioAte : Id, Nome, Ano -> {V,F,D}`

Foram também desenvolvidos os predicados que permitem inserção e remoção de conhecimento que serão descritos mais abaixo e o predicado **demo** que permite obter uma resposta de Verdadeiro, Falso ou Desconhecido para uma lista de predicados.

```
demo( [],[] ).
demo( [X|L],LR ) :-
    demoAux( X,R ),
    demo( L,[R|LR] ).
```

demoAux(Q,verdadeiro) :-

Q.

demoAux(Q,falso) :-

-Q.

demoAux(Q,desconhecido) :-

nao(Q),

nao(-Q).

Valores Nulos

Na construção do nosso caso prático de aplicação tivemos o cuidado de incluir pontos que dissessem respeito aos três tipos de valores nulos estudados previamente nas aulas.

O valor nulo incerto diz respeito a algo que é desconhecido, mas que futuramente poderá vir a conhecer-se. Por exemplo, no nosso enunciado, no ponto 5, é dito que a cor do automóvel 2 não se conhece. Eis como resolvemos essa questão:

cor(2,corx).

excecao(cor(F,P)) :-

cor(F,corx).

O valor nulo impreciso corresponde a algo que não se sabe ao certo qual a resposta mas há algumas possibilidades para a mesma. Por exemplo, no nosso enunciado, no ponto 8 é dito que o automóvel construído pelo Neves pode ser novo ou semi-novo. Esta questão foi resolvida da seguinte forma:

excecao(estado(2,novo)).

excecao(estado(2,seminovo)).

O valor nulo interdito diz respeito a algo que nunca será possível conhecer, ou seja, nunca será possível inserir esse conhecimento no sistema. No ponto 12 do nosso enunciado é dito que nunca será possível conhecer o ano de fabrico do automóvel 3. Segue-se a forma como resolvemos esta questão:

anofabrico(3,anox).

```
excecao( anofabrico( F,P ) ) :-  
    anofabrico( F,anox ).
```

```
nulo( anox ).
```

```
+anofabrico(F,P)::(solucoes((3,As),(anofabrico(3,As),nao(nulo(As))),S ),  
comprimento( S,N ), N == 0).
```

Inserção e Remoção de Conhecimento

Para possibilitar a inserção e remoção de conhecimento foram criados os predicados **evolução** e **remoção**.

O predicado remoção só permite a remoção de conhecimento que respeite os invariantes de remoção definidos.

% Extensão do predicado que permite a remoção de conhecimento: Termo -> {v, F}

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

```
remover(Termo) :-  
    retract(Termo).
```

O predicado evolução só permite a inserção de conhecimento que verifique os invariantes de inserção definidos por nós.

% Extensão do predicado que permite a evolucao do conhecimento

```
evolucao( Termo ) :- findall(Invariante, +Termo::Invariante, Lista),insercao(  
Termo),teste(Lista).
```

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

```
teste([]).  
teste([R|LR]) :-R,teste(LR).
```

Invariantes

Criamos vários invariantes para garantir a consistência do sistema: invariantes estruturais e invariantes referencias. Os primeiros foram criados para garantir que não é inserido conhecimento repetido no sistema, e são implementados da seguinte forma:

```
+automovelC( I,C )::(solucoes( (I,C),(automovelC( I,C )),S ),  
    comprimento( S,N ), N == 1).
```

Os segundos servem para garantir a coerência do sistema, por exemplo, não deixando que carro tenha mais do que um construtor. Da mesma forma não seria conveniente que tivesse mais do que um modelo, marca, estado ou ano de fabrico. No entanto pode ter mais do que uma cor e mais do que um registo de propriedade. Estes invariantes são implementados da seguinte forma:

```
+automovelC( I,C )::(solucoes( (Cs),(automovelC( I,Cs )),S ),  
    comprimento( S,N ), N =< 1).
```

Para além destes foi também necessário criar invariantes para os casos em que existem valor nulos interditos, não deixando que seja inserido conhecimento para esse valor. No nosso enunciado existe apenas um valor nulo desse tipo, correspondente ao facto de nunca se poder descobrir o ano de fabrico do automóvel 3. O invariante foi escrito da seguinte forma:

```
+anofabrico(F,P)::(solucoes((3,As),(anofabrico(3,As),nao(nulo(As))),S  
,comprimento( S,N ), N == 0 ).
```

Foi ainda criado um invariante que proíbe que seja removida a informação relativa ao construtor de um determinado automóvel quando este possui um ano de fabrico associado.

```
-automovelC(I,C) :: (-anofabrico(I,X)).
```

Interação Com o Sistema em Java

Foi-nos sugerido que desenvolvêssemos uma aplicação em Java para tornar o uso do programa mais intuitivo do que o *Prolog*. Para isso recorremos à biblioteca *Jasper* e criámos um programa com ligação ao *SICStus Prolog*.

```
*****
*****SRCR*****
*****Exercício 2*****
*****

1. Boletim de um automóvel.
2. Registos de propriedade de um automóvel.
3. Registos de propriedade de um automóvel desde um determinado ano.
4. Registos de propriedade de um automóvel até um determinado ano.
5. Automoveis de determinada cor.
6. Automoveis num determinado estado.
7. Automoveis de uma determinada marca.
8. Automoveis de um determinado modelo.
9. Automoveis de um determinado construtor.
10. Automoveis construídos num determinado ano.
11. Sair.
```

Figura 1 Menu da aplicação

Como é possível observar na Figura 1 o nosso programa em Java permite consultar várias informações sobre os automóveis existentes: permite consultar o seu boletim (contém todas as informações sobre o carro); os seus registos de propriedade com a opção de ser possível escolher um ano como margem para os registos que são mostrados; filtrar automóveis pelas diferentes características: cor, estado, marca, modelo, construtor e ano de fabrico.

Seguem-se alguns exemplos de *queries* efetuadas ao sistema e as respetivas respostas.

```
1
Insira id do automóvel:
5
Construtor:
Marca: {X=toyota}
Modelo: {X=grandeCarro}
Matrícula: {X=xzzy55}
Cor: {X=azul}
```

Figura 2 Resultado de pedir ao sistema o boletim de um automóvel

```

5
Insira cor:
azul
{X=3}
{X=5}
{X=7}

```

Figura 3 Resultado de pedir ao sistema todos os automóveis de cor azul

```

4
Insira id do automóvel:
5
Insira ano:
2015
Proprietários:
{X=manuel}
{X=rita}

```

Figura 4 Resultado de pedir ao sistema os proprietários de uma automóvel até um determinado ano

Análise de Resultados

Concluída a descrição da aplicação passamos para a confirmação de que a mesma se encontra funcionar corretamente. Para isso testamos vários predicados e comparamos com a informação do caso prático utilizado para verificar que as respostas são as esperadas.

- É suposto ser desconhecida a marca do automóvel com id 1. Logo, sempre que interrogamos sobre a sua marca a resposta deverá ser Desconhecido.

```

| ?- demoAux(automovelM(1,fiat),R).
R = desconhecido ? ■

```

Figura 5 Resultado da primeira query

- Da mesma forma é suposto que não se conheça a cor do automóvel 2.

```

| ?- demoAux(cor(2,verde),R).
R = desconhecido ?
yes
| ?- █

```

Figura 6 Resultado da segunda query

- O automóvel 3 poderá ter a matrícula xxzz33 ou xxzy33. Logo quando interrogado sobre estas duas matrículas deverá responder Desconhecido, quando interrogado sobre qualquer outra matrícula deverá responder Falso.

```

| ?- demoAux(matricula(3,xxzy33),R).
R = desconhecido ?
yes
| ?- █

| ?- demoAux(matricula(3,xxff44),R).
R = falso ?
yes

```

Figura 7 Resultado da terceira query

- Nunca poderá ser inserida informação sobre o ano de fabrico do automóvel 3, logo quando tentamos o predicado evolução não deverá permitir.

```

| ?- evolucao(anofabrico(3,2000)).
no
| ?- █

```

Figura 8 Resultado da quarta query

- O estado do automóvel 2 é pode ser novo ou seminovo, logo para estes dois valores a resposta deverá ser Desconhecido e para os restantes deverá ser Falso.

```

| ?- demoAux(estado(2,novo),R).
R = desconhecido ?
yes
| ?- demoAux(estado(2,seminovo),R).
R = desconhecido ?
yes
| ?- demoAux(estado(2,velho),R).
R = falso ?
yes
| ?- █

```

Figura 9 Resultado da quinta query

Conclusões e Sugestões

Com este trabalho conseguimos perceber melhor o conceito de conhecimento imperfeito bem como o significado de cada um dos valores nulos que tínhamos estudado, bem como a utilidade de tudo isto para situações da vida real.

Gostaríamos de ter conseguido um melhor resultado no que toca à parte realizada em Java pois o objetivo era ter uma aplicação que produzisse os mesmos resultados que obtemos no *Prolog* mas que funcionasse de forma mais intuitiva. No entanto não conseguimos implementar tudo o que desejávamos o que fez com que a aplicação ficasse um pouco limitada e não fosse possível através dela responder às questões que tínhamos proposto no enunciado que criamos. Por esse motivo decidimos no capítulo de análise de resultados utilizar os resultados obtidos no *Prolog* e utilizar a aplicação em Java apenas como um “extra”.

Foi interessante perceber que o *Prolog* pode estar ligado desta forma ao Java e as coisas que é possível fazer com esta ligação.