**Universidade do Minho**

**Exercício 2**

Licenciatura em Engenharia Informática

Sistemas de Representação de Conhecimento e Raciocínio

2ºSemestre (2014/2015)

67673 André Geraldes

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 Raciocino e consiste no desenvolvimento de um sistema de representação de conhecimento imperfeito que caracterize um universo de comércio automóvel.

Índice

[Resumo 2](#_Toc415000598)

[Índice de Figuras 4](#_Toc415000600)

[Introdução 5](#_Toc415000601)

[Preliminares 6](#_Toc415000602)

[Descrição do Trabalho 7](#_Toc415000603)

[Caso Prático de Aplicação 7](#_Toc415000604)

[Desenvolvimento dos Predicados 7](#_Toc415000605)

[Inserção e Remoção de Conhecimento 13](#_Toc415000606)

[Análise de Resultados 14](#_Toc415000607)

[Conclusões e Sugestões 16](#_Toc415000608)

Índice de Figuras

[Figura 1 - Árvore genealógica do caso prático utilizado 7](#_Toc415000377)

[Figura 2 - Resultados obtidos 15](#_Toc415000378)

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, possuirmos 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 raciocino 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. 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 o estado do seu carro é novo ou semi-novo.
9. O carro do 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 é 2010.
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.
17. A Sara nunca irá saber quem foram os proprietários do seu automóvel entre 1994 e 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 críamos 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 proprietarios: Id,Resultados -> {V,F}
* predicado proprietarioDesde : Id,Nome,Ano -> {V,F,D}
* predicado proprietariosDesde : Id,Ano,Resultado -> {V,F}
* predicado proprietarioAte : Id,Nome,Ano -> {V,F,D}
* predicado proprietariosAte : Id,Ano,Resultado -> {V,F}

Resposta ao enunciado proposto

Passamos então a demonstrar a forma como demos resposta ao enunciado por nós construído utilizando os predicados identificados anteriormente.

O Xico é o construtor do automóvel com identificador 1.

automovelC(1,xico).

Sabe-se que o modelo do carro é grandeCarro.

automovelm(1,grandeCarro).

Mas desconhece-se a sua marca.

automovelM( 1,marcax ).

excecao( automovelM( F,P ) ) :-

automovelM( F,marcax ).

O automóvel, cujo id é 2, é da marca Toyota do modelo pequenoCarro.

automovelM(2,toyota).

automovelm(2,pequenoCarro).

O construtor deste carro é o Neves.

automovelC(2,neves).

Inserção e Remoção de Conhecimento

Para possibilitar a inserção e remoção de conhecimento foram criados os predicados **evolução** e **remoção.** O predicado remoção não é mais do que uma chamada do ***retract.***

% Extensão do predicado que permite a remoção do conhecimento

remocao(Termo) :- retract(Termo).

O predicado evolução só permite a inserção de conhecimento que verifique os invariantes 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).

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 ).

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.

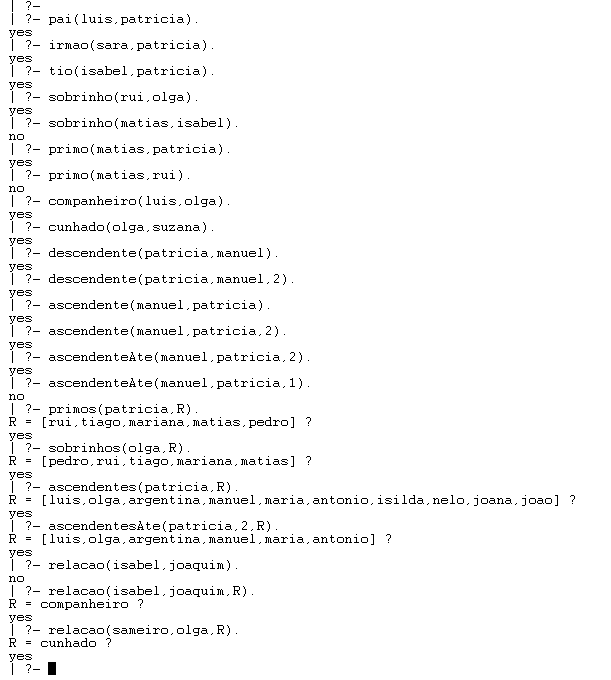


Figura 2 - Resultados obtidos

Conclusões e Sugestões

Com este trabalho foi possível observarmos como os sistemas de conhecimento e raciocínio podem ser úteis e interessante quando aplicados a casos práticos. Após a sua realização todos nós percebemos e apreciamos mais o ***PROLOG.***

Consideramos que produzimos um trabalho bastante satisfatório tendo implementado todas as funcionalidades exigidas e ainda alguns pormenores que não eram pedidos.