

## Grupo: Caio Buarque de Gusmão e José Gabriel Santos Nascimento

### Descrição do Problema:

A canção "I'm my Own Grandpa" (eu sou meu próprio avô), relata a história de uma família nada convencional:

<p>I'm my Own Grandpa By Dwight Latham and Moe Jaffe Performed by Ray Stevens</p> <p>Many, many years ago When I was twenty-three I was married to a widow Who was pretty as can be This widow had a grown-up daughter Who had hair of red My father fell in love with her And soon the two were wed</p> <p>This made my dad my son-in-law And it really changed my life For my daughter was my mother 'Cause she was my father's wife To complicate the matter Even though it brought me joy I soon became the father Of a bouncing baby boy</p> <p>My little baby then became A brother-in-law to Dad And so became my uncle Though it made me very sad For if he was my uncle Then that also made him brother Of the widow's grown-up daughter</p>	<p>Eu sou meu próprio avô Por Dwight Latham e Moe Jaffe Interpretado por Ray Stevens</p> <p>Muitos, muitos anos atrás Quando eu tinha vinte e três anos Eu era casado com uma viúva Que era tão bonita quanto possível Esta viúva tinha uma filha crescida Que tinha cabelos ruivos Meu pai se apaixonou por ela E logo os dois se casaram</p> <p>Isso fez de meu pai meu filho<sup>3</sup> E de fato mudou minha vida Pois minha filha era minha mãe Porque ela era a esposa de meu pai. Para complicar o assunto Embora isso realmente me tenha trazido alegria Eu logo me tornei o pai De um menino saltitante</p> <p>Este pequeno bebê então se tornou Um irmão do papai E assim se tornou meu tio Embora isso me tenha entristecido muito Pois se ele era meu tio Então isso também o fez irmão Da filha adulta da viúva</p>
<p>Who is of course my step-mother</p> <p>My father's wife then had a son Who kept them on the run And he became my grandchild For he was my daughter's son My wife is now my mother's mother And it makes me blue Because although she is my wife She's my grandmother too</p> <p>Now if my wife is my grandmother Then I'm her grandchild And every time I think of it It nearly drives me wild For now I have become The strangest case you ever saw As husband of my grandmother I am my own grandpa</p> <p>Chorus I'm my own grandpa I'm my own grandpa It sounds funny I know But it really is so Oh, I'm my own grandpa</p>	<p>Que, é claro, é minha madrasta</p> <p>A esposa de meu pai então teve um filho Que os manteve na correria E ele se tornou meu neto Pois ele era filho de minha filha Minha esposa é agora a mãe de minha mãe. E isso me deixa triste Porque embora ela seja minha esposa Ela é minha avó também</p> <p>Agora, se minha esposa é minha avó Então eu sou seu neto E toda vez que penso nisso Isso quase me deixa louco Porque agora eu me tornei O caso mais estranho que você já viu Como marido de minha avó Eu sou meu próprio avô</p> <p>Refrão Eu sou meu próprio avô Eu sou meu próprio avô Parece engraçado, eu sei. Mas realmente é assim Oh, eu sou meu próprio avô</p>

Para facilitar a análise de tais relações familiares, fizemos um código em prolog a partir de alguns fatos iniciais:

```
child(redhair, widow).
child(i, dad).
child(onrun, dad).
child(baby, i).
male(i).
male(dad).
male(onrun).
male(baby).
female(redhair).
female(widow).
spouse(i, widow).
spouse(widow, i).
spouse(dad, redhair).
spouse(redhair, dad).
```

child(X,Y), significa dizer que "X" é filho de "Y";  
male(X) significa dizer que "X" é do sexo masculino;  
female(X) significa dizer que "X" é do sexo feminino;  
spouse(X,Y) significa dizer que "X" é casado com "Y";  
"i" faz referência ao narrador da canção;  
"widow" é a viúva esposa do narrador;  
"dad" é o pai do narrador;  
"redhair" é a filha de "widow";  
"baby" é o filho do narrador e ;  
"onrun" é o filho de "dad";

Tínhamos de escrever algumas regras para responder "true" para a seguinte sequência:

```
son(dad, i), daughter(redhair, i), mother(redhair, i),
brother(baby, dad), uncle(baby, i), brother(baby,
redhair), mother(redhair, i), grandchild(onrun, i), mother(widow,
redhair), grandmother(widow, i), grandchild(i,
widow), grandfather(i, i).
```

Nosso código cumpriu esse requisito como mostra na imagem a seguir.

```

SWI-Prolog (AMD64, Multi-threaded, version 8.4.2)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/users/pichau/onedrive/documentos/prolog/fatosiniciais compiled 0.00 sec, -2 clauses
?- son(dad, i), daughter(redhair, i), mother(redhair, i),
   brother(baby, dad), uncle(baby, i), brother(baby,
   redhair), mother(redhair, i), grandchild(onrun, i), mother(widow,
   redhair), grandmother(widow, i), grandchild(i,
   | | widow). grandfather(i, i).
true .

```

## Destaques da Implementação:

### Uncles:

O predicado “uncles(Resp)” foi feito com um predicado embutido de prolog, o setof/3.

```

uncles(X) :-
    setof(Y, uncle(Y,X), X0), write(X0).

```

Este predicado precisa de 3 argumentos: objeto(Y), objetivo(uncle(Y,X) e lista(X0). Ele vai produzir uma lista X0, que estará ordenada e sem duplicados, de todos os objetos Y que satisfazem o uncle(Y,X), nesse caso ele vai conferir se Y é tio de X, se for, vai estar na lista X0. No fim, o “write(X0)” vai escrever a lista.

**Exemplo de implementação:** lista de tios de “i”: escrevemos “uncles(i)”, então vai chamar o predicado:

```

uncles(X) :-
    setof(Y, uncle(Y,X), X0), write(X0).

```

onde X vai ser “i” e ele vai procurar valores de Y que satisfaçam o uncle(Y,i), ou seja, vai buscar os tios de “i” com o predicado:

```

uncle(X, Y) :-
    mother(M, Y),
    brother(X, M),
    male(X);

    father(M, Y),
    brother(X, M),
    male(X).

```

esse predicado vai verificar se Y é do sexo masculino e se é irmão do pai ou da mãe de “i”, no caso, e tais valores de Y vão ficar na lista X0. No fim, quando tudo for verificado e chegar no “write(X0)” o resultado será:

```

?- uncles(i).
[baby, dad]

```

Son e Daughter:

Vale ressaltar que estamos levando em conta as relações familiares "in-law", ou seja, madrasta também é mãe, padrasto também é pai, enteado é filho, genro é filho, nora é filha.

<b>son</b> (X,Y) :- child(X,Y) , male(X) ;  child(Z,R) , spouse(Y,R) , spouse(X,Z) , male(X) ;  child(H,Y) , spouse(H,X) , male(X) ;  child(X,Z) , spouse(Z,Y) , male(X) .	<b>daughter</b> (X,Y) :- child(X,Y) , female(X) ;  child(Z,R) , spouse(Y,R) , spouse(X,Z) , female(X) ;  child(H,Y) , spouse(H,X) , female(X) ;  child(X,Z) , spouse(Z,Y) , female(X) .
---	--

Podemos ver que se X for filho de Y e X for homem, X é filho de Y. Se X é casado com o filho(a) de da esposa/marido de Y (X for genro da esposa ou marido de Y) e X for homem, X é filho de Y. Se X for casado com o filho de Y (X genro de Y) e X for homem, X é filho de Y. Se X for filho da esposa/marido de Y (X enteado de Y) e X for homem, X é filho de Y. Para daughter é o mesmo pensamento, porém é uma mulher.

## Código passado em Lógica dos Predicados:

---

### Fatos

male(x) => x é homem

female(x) => x é mulher

spouse(x, y) => x é casado com y

child(x, y) => x é filho(a) de y

---

## Regras

$$\forall x \forall y \forall z \forall r \forall h \{ \text{son}(x, y) \rightarrow [\text{child}(x, y) \wedge \text{male}(x)] \vee [\text{child}(z, r) \wedge \text{spouse}(y, r) \wedge \text{spouse}(x, z) \wedge \text{male}(x)] \vee [\text{child}(h, y) \wedge \text{spouse}(h, x) \wedge \text{male}(x)] \vee [\text{child}(x, z) \wedge \text{spouse}(z, y) \wedge \text{male}(x)] \}$$

$$\forall x \forall y \forall z \forall r \forall h \{ \text{daughter}(x, y) \rightarrow [\text{child}(x, y) \wedge \text{female}(x)] \vee [\text{child}(z, r) \wedge \text{spouse}(y, r) \wedge \text{spouse}(x, z) \wedge \text{female}(x)] \vee [\text{child}(h, y) \wedge \text{spouse}(h, x) \wedge \text{female}(x)] \vee [\text{child}(x, z) \wedge \text{spouse}(z, y) \wedge \text{female}(x)] \}$$

$$\forall x \forall y \{ \text{mother}(x, y) \rightarrow [\text{son}(y, x) \wedge \text{female}(x)] \vee [\text{daughter}(y, x) \wedge \text{female}(x)] \}$$

$$\forall x \forall y \{ \text{father}(x, y) \rightarrow [\text{son}(y, x) \wedge \text{male}(x)] \vee [\text{daughter}(y, x) \wedge \text{male}(x)] \}$$

$$\forall x \forall y \forall p \forall m \{ \text{brother}(x, y) \rightarrow [\text{father}(p, x) \wedge \text{father}(p, y) \wedge \text{male}(x)] \vee [\text{mother}(m, x) \wedge \text{mother}(m, y) \wedge \text{male}(x)] \}$$

$$\forall x \forall z \forall y \forall v \{ \text{grandchild}(x, y) \rightarrow [\text{mother}(y, z) \wedge \text{mother}(z, x)] \vee [\text{father}(v, x) \wedge \text{mother}(y, v)] \vee [\text{father}(y, z) \wedge \text{mother}(z, x)] \vee [\text{father}(y, v) \wedge \text{father}(v, x)] \}$$

$$\forall x \forall y \{ \text{grandmother}(x, y) \rightarrow \text{grandchild}(y, x) \wedge \text{female}(x) \}$$

$$\forall x \forall y \{ \text{grandfather}(x, y) \rightarrow \text{grandchild}(y, x) \wedge \text{male}(x) \}$$

$$\forall x \forall y \forall m \{ \text{uncle}(x, y) \rightarrow [\text{mother}(m, y) \wedge \text{brother}(x, m) \wedge \text{male}(x)] \vee [\text{father}(m, y) \wedge \text{brother}(x, m) \wedge \text{male}(x)] \}$$

## CONCLUSÕES:

Através dos conhecimentos de lógica junto ao uso do sistema de prolog, o grupo foi capaz de criar uma solução para um problema que poderia ser resolvido de maneira manual mas com dificuldade e falta de eficiência.

Dessa forma, com base nas complexas relações familiares que o problema apresentava, foi possível desenvolver um programa através da utilização de regras universais que continham as condições necessárias para encontrar os parentescos buscados. Com isso é mostrada a importância de se compreender os conceitos vistos no conteúdo de "Lógica Formal Moderna" para resolver problemas de lógica com complexidade semelhante ou superior apresentada no caso estudado.

Dentro da tradução do programa para Lógica dos Predicados é possível perceber a constante presença do uso de certos conectivo. Isso porque cada um vai exercer um papel, o condicional serve como indicador da condição necessária para

para determinar as novas regras que serão usadas no programa, a conjunção para tratar o conjunto de regras e a disjunção para lidar com as diferentes situações que podem ocorrer no problema.

## **BIBLIOGRAFIA:**

<https://www.educba.com/prolog-setof/>

**Uma Introdução a Lógica Formal Moderna, Sérgio Queiroz, adaptado de “A Modern Formal Logic Primer” de Paul Teller.**