



Instituto Superior de Engenharia de Lisboa
ADEETC
Licenciatura em Engenharia Informática e de Computadores
(1º Sem 2019/2020)

Lógica e Computação – Primeiro trabalho

Docente: Walter Vieira

1ª parte – Cálculo de proposições

1. Considere as seguintes premissas:

F1: “Se chover, trago o chapéu de chuva”

F2: “Se fizer sol, não trago o chapéu de chuva”

Mostre se se pode concluir:

- a) Não é possível chover e fazer sol em simultâneo.
- b) Chover é equivalente a não fazer sol.

2. Considere as seguintes frases:

F1: “Passa quem, e apenas quem, estudar”,

F2: “Se o José passar, ganha um automóvel”,

G: “Se o José ..., então não estuda”.

Indique, justificando, três substituições possíveis para a parte de G indicada por “...” que fazem com que $\{F1, F2\} \models G$.

3. Considere as seguintes formulas:

F1: “Se chover, trago o chapéu de chuva”

F2: “Se chover, não vou à praia”

F3: “Se fizer sol, fico satisfeito”

F4: “Faz sol”

Mostre, através da árvore de resolução, se se pode concluir:

G: “Se for à praia, não chove”

4. Dois alunos observam um indivíduo junto ao quadro e afirmam:

Aluno 1:

“Ele ou é aluno, ou é professor”

“Ele ou não é aluno, ou finge que é professor”

Aluno 2:

“ele é aluno se e só se não for professor”

“ele finge que é professor se e só se for aluno”

Admitindo que o conhecimento de ambos os alunos é correto, diga, justificando, qual dos dois tem o conhecimento mais profundo.

Nota: Se pretender resolver este problema usando o LogIs, deve colocar as opções Maximum depth e Depth increment com o valor 10 e selecionar a opção Framed literals.

2ª parte – Cálculo de predicados

1. Active a opção “trace” do LOGIS.

Introduza o seguinte conjunto de premissas:

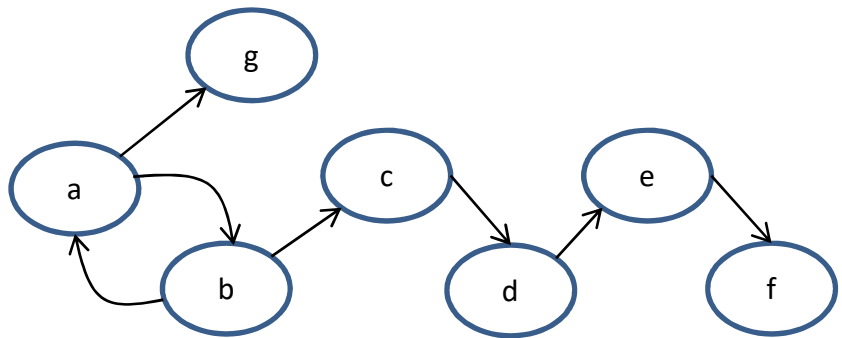
$\text{rico}(\text{pai}(\text{pai}(\text{rita})))$.
 $(\forall x)(\text{rico}(\text{pai}(x)) \rightarrow \text{rico}(x))$.

- a) Construa a árvore de resolução para o objetivo $\text{rico}(\text{rita})$ e confirme-a com o LogIs.
- b) Elimine a primeira premissa, repita o exercício da alínea anterior e justifique o que observa.

2. Considere o mapa de ligações entre cidades mostrado na figura seguinte:

O conjunto de premissas em lógica de 1ª ordem seguinte modela este problema:

$\text{connects}(a,g)$.
 $\text{connects}(a,b)$.
 $\text{connects}(b,a)$.
 $\text{connects}(b,c)$.
 $\text{connects}(c,d)$.
 $\text{connects}(d,e)$.
 $\text{connects}(e,f)$.



$(\forall x)(\forall y)(\text{connects}(x,y) \rightarrow \text{route}(x,y,l(x,l(y,\text{nil}))))$.
 $(\forall x)(\forall y)(\forall z)(\forall w)(\text{connects}(x,y) \wedge \text{route}(y,z,w) \rightarrow \text{route}(x,z,l(x,w)))$.

Introduza estas premissas no LogIs

- a) Defina as opções Maximum depth e Depth increment do LogIs com o valor 20 e peça ao LogIs para provar os seguintes objetivos:

- i. $(\exists x)(\text{route}(c,f,x))$.
- ii. $(\exists x)(\text{route}(a,d,x))$.
- iii. $(\exists x)(\text{route}(a,c,x))$.

- b) Redefina a opção Depth increment do LogIs com o valor 1 e repita o objetivos da alínea a).

Compare e justifique os resultados obtidos nas alíneas a) e b).

- c) Com a opção Depth increment igual a 1 Interroque o LogIs sobre a existência de caminhos que partam de **a** e tenham um destino qualquer. Anote as primeiras 8 soluções.
- d) Redefina a opção Incremento do LogIs com o valor 20 e repita o objetivo da alínea c).

Compare e justifique os resultados obtidos nas alíneas c) e d).

3. Considere a definição de listas seguinte:

Lista vazia \equiv nil

Lista com cabeça c e resto r (uma lista) \equiv l(c,r)

Defina as opções Maximum depth e Depth increment do LogIs com o valor 50.

Perceba e introduza no LogIs as premissas existentes no ficheiro **listsLogIs.txt** (o predicado $\&add(x,y,z)$ é um predicado “built in” do LogIs que sucede se z for o resultado da soma de x com y e a função $\&add(x,y)$ é expandida para o predicado $\&add(x,y,z)$, como se pode ver na forma clausal). Antes de resolver as alíneas seguintes, veja as premissas na forma clausal e anote as cláusulas.

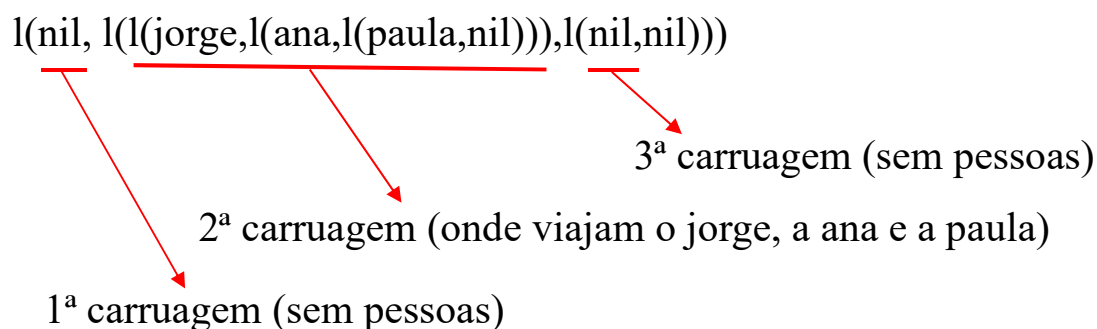
- Interroque o LogIs quanto à existência de listas (x) de dimensão 3 que tenham como membros listas de dimensão 1 cujo único membro também é membro da lista x.
- Construa a árvore de resolução para o objetivo **#(l,invert(l,l(a,l(b,nil))))**.
- Realize o predicado **palindromo(l)** que sucede se a sequência de elementos existentes na lista l for um palíndromo.
- Interroque o LogIs quanto à existência de listas palíndromo das quais “a” é membro.
- Interroque o LogIs quanto à existência de listas palíndromo de dimensão 3, das quais a e b sejam membros.

4. Considere a seguinte definição de uma lista:

Átomo nil – lista vazia.

Função l(c,r) – lista com cabeça c e resto r.

Usando esta notação, pode representar-se um comboio por uma lista de carruagens, onde cada carruagem é, por sua vez, uma lista com os nomes das pessoas que nela viajam, como se mostra a seguir:



- Construa o predicado **procurar(comboio, nome, numeroCarruagem)** que sucede se **numeroCarruagem** for o número da carruagem (a primeira é a número 0) onde viaja a pessoa de nome **nome**, no comboio **comboio**.

- b) Construa a árvore de resolução que permite saber em que carruagem viaja a **paula** no combóio representado por $l(\text{nil}, l(l(\text{jorge}, l(\text{paula}, l(\text{ana}, \text{nil}))), l(\text{nil}, \text{nil})))$.

Admita a existência do predicado “built-in” $\&\text{add}(x,y,z)$ que sucede se a soma de x com y for z .

3ª parte – Prolog (iniciação)

1. Escreva as cláusulas Prolog correspondentes às seguintes frases (como premissas):

F₁: “Todos os animais morrem”.
F₂: “Todos são animais”.
F₃: “Os gatos comem ratos”.
F₄: “O Tareco e o Farrusco são gatos e o Pluto é cão”.
F₅: “Os gatos miam e arranham”.
F₆: “Os cães mordem se estiverem de mau humor ou se forem maus”.

Escreva as cláusulas correspondentes aos objetivos seguintes e explore o programa obtido com as premissas anteriores usando esses objetivos:

G₁: “O Tareco e o Farrusco arranham e comem ratos”.
G₂: “Ou o Pluto morde, ou o Farrusco arranha”.
G₃: “Existem ratos que são comidos por gatos”.
G₄: “Todos morrem”

-
2. Considere o seguinte programa Prolog (*fornecido no ficheiro Parte3.pl*):

```
goodChoice(X,Y) :- select(X,Y),capable(Y).
select(X,Y) :- likes(X,Y).
goodChoice1(X,Y) :- select1(X,Y),capable(Y).
select1(X,Y) :- likes(X,Y),!.
likes(ana,rui).
likes(ana,pedro).
capable(pedro).
```

Explore-o com os objetivos **goodChoice(ana,X)**, **goodChoice1(ana,X)** e **goodChoice1(ana,pedro)** e justifique os resultados obtidos.

-
3. Considere os seguintes predicados Prolog (*fornecidos no ficheiro Part3.pl*):

```
fact(0,1).
fact(N,N*X) :- fact(N-1,X).

fact1(0,1).
fact1(N,X) :- N > 0, N1 is N-1, fact1(N1,Y), X is N*Y.

fact2(0,1).
fact2(N,X) :- fact2(N1,Y), N is N1+1, X is N*Y.

fact3(0,suc(0)).
```

$\text{fact3}(\text{suc}(\text{N}), \text{X}) \text{ :- fact3}(\text{N}, \text{Y}), \text{mult}(\text{Y}, \text{suc}(\text{N}), \text{X}).$

$\text{mult}(\text{X}, 0, 0) \text{ :- !}.$

$\text{mult}(0, \text{X}, 0) \text{ :- !}.$

$\text{mult}(\text{X}, \text{suc}(\text{Y}), \text{M}) \text{ :- mult}(\text{X}, \text{Y}, \text{M1}), \text{sum}(\text{X}, \text{M1}, \text{M}).$

$\text{sum}(0, \text{X}, \text{X}) \text{ :- !}.$

$\text{sum}(\text{X}, 0, \text{X}) \text{ :- !}.$

$\text{sum}(\text{X}, \text{suc}(\text{Y}), \text{suc}(\text{Z})) \text{ :- sum}(\text{X}, \text{Y}, \text{Z}).$

Justifique os comportamentos que se obtêm com os objetivos:

- a) (1.^a solução)
 - a. **fact(0,X),**
 - b. **fact1(0,X),**
 - c. **fact2(0,X)**
 - d. **fact3(0,X).**
- b) (1.^a solução)
 - a. **fact(3,X),**
 - b. **fact1(3,X),**
 - c. **fact2(3,X),**
 - d. **fact3(suc(suc(suc(0))),X).**
- c) (1.^a solução)
 - a. **fact(X,6),**
 - b. **fact1(X,6),**
 - c. **fact2(X,6),**
 - d. **fact3(X,suc(suc(suc(suc(suc(suc(0))))))).**
- d) (4 primeiras soluções)
 - a. **fact(X,Y),**
 - b. **fact1(X,Y),**
 - c. **fact2(X,Y),**
 - d. **fact3(X,Y)**

Prazo de entrega:

Data limite para entrega (no moodle): 2019/11/08 às 23:50 horas.

(mas aconselham-se os alunos a fazerem o que conseguirem o mais cedo possível)