Aula Prática 1

Representação de Conhecimento e Funcionamento do Prolog

Objetivos:
- Introdução à linguagem Prolog
- Representação de conhecimento em Prolog – Factos e Regras
- Utilização do interpretador Prolog – Questões e Variáveis
- Funcionamento do Prolog – Backtracking

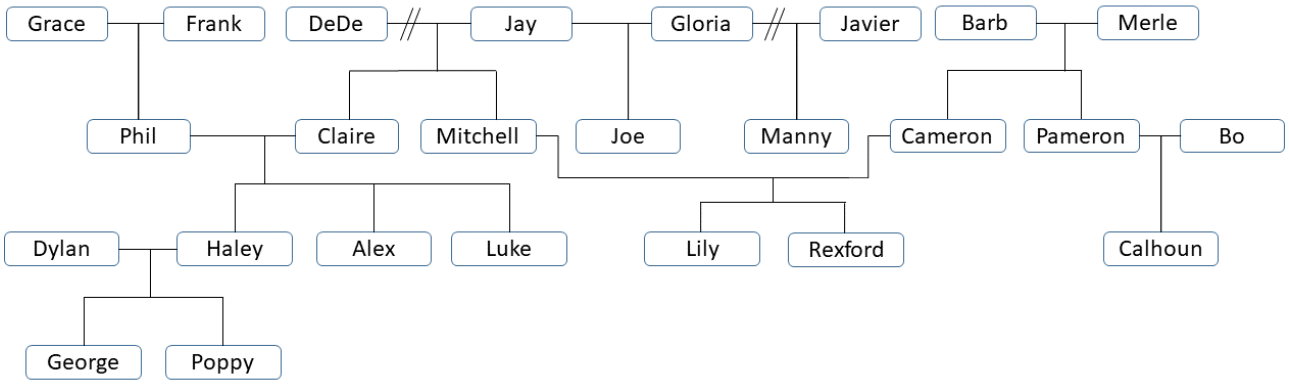
0. Software

Se ainda não o fez, instale o *SICStus Prolog* (ou outro interpretador, como o *SWI Prolog*, *YAP*, ou outro). Opcionalmente pode instalar também o SPIDER (*SICStus Prolog IDE for Eclipse*).

1. Relações Familiares

a) Represente em Prolog a informação presente na árvore genealógica abaixo recorrendo apenas aos predicados *male/1*, *female/1* e *parent/2*.

Nota: *parent(a, b)* é interpretado como '*a* é progenitor de *b*'.



b) Usando o interpretador, obtenha respostas para as seguintes questões:

i. A Haley é do sexo feminino?
ii. O Gil é do sexo masculino?
iii. O Frank é progenitor do Phil?
iv. Quem são os pais da Claire?
v. Quem são os filhos da Gloria?
vi. Quem são os netos do Jay?
vii. Quem são os avós da Lily?
viii. A Alex tem filhos?
ix. Quem é filho do Jay, mas não da Gloria?

c) Escreva regras que permitam definir relações familiares mais complexas, como *father/2*, *grandparent/2*, *grandmother/2*, *siblings/2*, *halfSiblings/2*, *cousins/2*, *uncle/2*, ou outras.

d) Usando o interpretador, faça questões para testar as relações definidas na alínea anterior, como 'Haley e Lily são primas?', 'quem é o pai de Luke?', 'quem é tio de Lily?', 'quem é avó?', ou listar todos os pares de irmãos, ou de meios-irmãos.

e) Que predicados utilizaria para representar informação de casamentos e divórcios? Como representaria o casamento de Jay com Gloria em 2008, ou o casamento de Jay com Dede em 1968 e respetivo divórcio em 2003?

2. Professores e Alunos

a) Represente informação relativa a unidades curriculares (UCs), professores e estudantes, de acordo com a tabela abaixo, usando os predicados *leciona/2* e *frequenta/2*, em que o primeiro argumento de cada um representa a UC lecionada ou frequentada.

Algoritmos	Bases de Dados	Compiladores	Estatística	Redes
Prof: Adalberto	Prof: Bernardete	Prof: Capitolino	Prof: Diógenes	Prof: Ermelinda
Alunos: Alberto, Bruna, Cristina, Diogo, Eduarda	Alunos: António, Bruno, Cristina, Duarte, Eduardo	Alunos: Alberto, Bernardo, Clara, Diana, Eurico	Alunos: António, Bruna, Cláudio, Duarte, Eva	Alunos: Álvaro, Beatriz, Cláudio, Diana, Eduardo

b) Use o interpretador para responder às seguintes questões:
   i. Que UC leciona o Diógenes?
   ii. A Felismina leciona alguma UC?
   iii. Que UCs frequenta o Cláudio?
   iv. O Dalmindo frequenta alguma UC?
   v. A Eduarda é aluna da Bernardete?
   vi. O Alberto e o Álvaro frequentam alguma UC em comum?

c) Escreva regras em Prolog que permitam responder às seguintes questões:
   i. X é aluno do professor Y?
   ii. Quem são os alunos do professor X?
   iii. Quem são os professores do aluno X?
   iv. Quem é ao mesmo tempo aluno do professor X e do professor Y?
   v. Quem é colega de quem? (dois alunos são colegas se frequentarem pelo menos uma UC em comum; dois docentes são colegas)
   vi. Quem são os alunos que frequentam mais de 1 UC?

3. Red Bull Air Race

a) Represente o seguinte conhecimento em Prolog:
   - Lamb, Besenyei, Chambliss, MacLean, Mangold, Jones e Bonhomme são pilotos;
   - Lamb é da equipa Breitling; Besenyei e Chambliss da Red Bull; MacLean da Mediterranean Racing Team; Mangold da Cobra; Jones e Bonhomme da Matador;
   - O avião de Lamb é um MX2 e o de Besenyei, Chambliss, MacLean, Mangold, Jones e Bonhomme é um Edge540;
   - Istanbul, Budapest e Porto são circuitos;
   - Jones venceu no Porto; Mangold venceu em Budapest e em Istanbul;
   - Istanbul tem 9 gates; Budapest tem 6 gates; Porto tem 5 gates;
   - Uma equipa ganha uma corrida quando um dos seus pilotos vence essa corrida.

b) Escreva as seguintes perguntas em Prolog:
   i. Quem venceu a corrida no Porto?
   ii. Qual a equipa que ganhou no Porto?
   iii. Que circuitos têm mais de 8 gates?
   iv. Que pilotos não pilotam um Edge540?
   v. Quais os pilotos que venceram mais de um circuito?
   vi. Qual o avião pilotado pelo piloto que venceu a corrida no Porto?

4. Programação e Erros

Um estudante acostumado a usar linguagens imperativas está a desenvolver um compilador em Prolog. Uma das tarefas consiste em traduzir um código de erro para uma descrição em português. O código que ele criou foi:

```
traduz(Codigo, Significado):-
      Codigo = 1,
      Significado = 'Integer Overflow'.
traduz(Codigo, Significado) :-
      Codigo = 2,
      Significado = 'Divisao por zero'.
traduz(Codigo, Significado) :-
      Codigo = 3,
      Significado = 'ID Desconhecido'.
```

Com sabe, esta não é uma forma apropriada de programar em Prolog. Melhore este código.

5. Cargos e Chefes

Considere a seguinte Base de Factos Prolog, com os predicados *cargo/2* e *chefiado_por/2*:

```
cargo(tecnico, eleuterio).
cargo(tecnico, juvenaldo).
cargo(analista, leonilde).
cargo(analista, marciliano).
cargo(engenheiro, osvaldo).
cargo(engenheiro, porfirio).
cargo(engenheiro, reginaldo).
cargo(supervisor, sisnando).
cargo(supervisor_chefe, gertrudes).
cargo(secretaria_exec, felismina).
cargo(diretor, asdrubal).
chefiado_por(tecnico, engenheiro).
chefiado_por(engenheiro, supervisor).
chefiado_por(analista, supervisor).
chefiado_por(supervisor, supervisor_chefe).
chefiado_por(supervisor_chefe, diretor).
chefiado_por(secretaria_exec, diretor).
```

a) Sem usar o computador, descreva em linguagem natural as seguintes interrogações:

```
 i.   | ?- chefiado_por(analista, X), cargo(X, sisnando).
 ii.  | ?- chefiado_por(tecnico, X), chefiado_por(X, Y).
 iii. | ?- cargo(J, P), chefiado_por(J, supervisor).
 iv.  | ?- chefiado_por(P, diretor), \+(cargo(P, felismina)).
```

b) Sem utilizar o computador, indique qual seria a primeira resposta encontrada pelo Prolog para cada uma das interrogações acima.

c) Escreva regras que permitam responder às seguintes questões:
   i. A pessoa X é chefe da pessoa Y?
   ii. As pessoas X e Y são chefiadas por pessoas com o mesmo cargo?
   iii. Quais os cargos que não são responsáveis por outros cargos?
   iv. Quem são as pessoas que não são chefiadas por ninguém?

6. Backtracking e Árvore de Pesquisa

Considere os seguintes factos e regras:

```
pairs(X, Y):- d(X), q(Y).
pairs(X, X):- u(X).
u(1).
d(2).
d(4).
q(4).
q(16).
```

a) Sem usar o interpretador Prolog, esboce uma árvore de pesquisa e indique todas as soluções obtidas para a consulta `| ?- pairs(X, Y).`

7. Funcionamento do Backtracking

Considere os seguintes factos e regras:

```
a(a1, 1).
a(A2, 2).
a(a3, N).
b(1, b1).
b(2, B2).
b(N, b3).
c(X, Y):- a(X, Z), b(Z, Y).
d(X, Y):- a(X, Z), b(Y, Z).
d(X, Y):- a(Z, X), b(Z, Y).
```

b) Sem usar o Prolog, indique quais serão as respostas às seguintes perguntas:

```
 i.   a(A, 2).
ii.   b(A, foobar).
iii.  c(A, b3).
iv.   c(A, B).
 v.   d(A, B).
```

c) Confirme as suas respostas com o rastreio (*trace*) do Prolog.