

Tema: Introdução à programação lógica.

Objetivos: Desenvolver bases de conhecimento para modelar problemas simples. Implementar mecanismos de raciocínio lógico sobre essas bases de conhecimento.

Exercício 1

Considere as relações a árvore genealógica que se apresenta na Figura 1.

- 1.1 Relativamente a esta árvore, que conhecimento é que ela codifica? Dê exemplo de relacionamentos e factos nela codificados.
- 1.2 Implemente, em Prolog, o conhecimento assinalado utilizando o mínimo de código possível. Faça queries à base de conhecimento e verifique a sua correção.

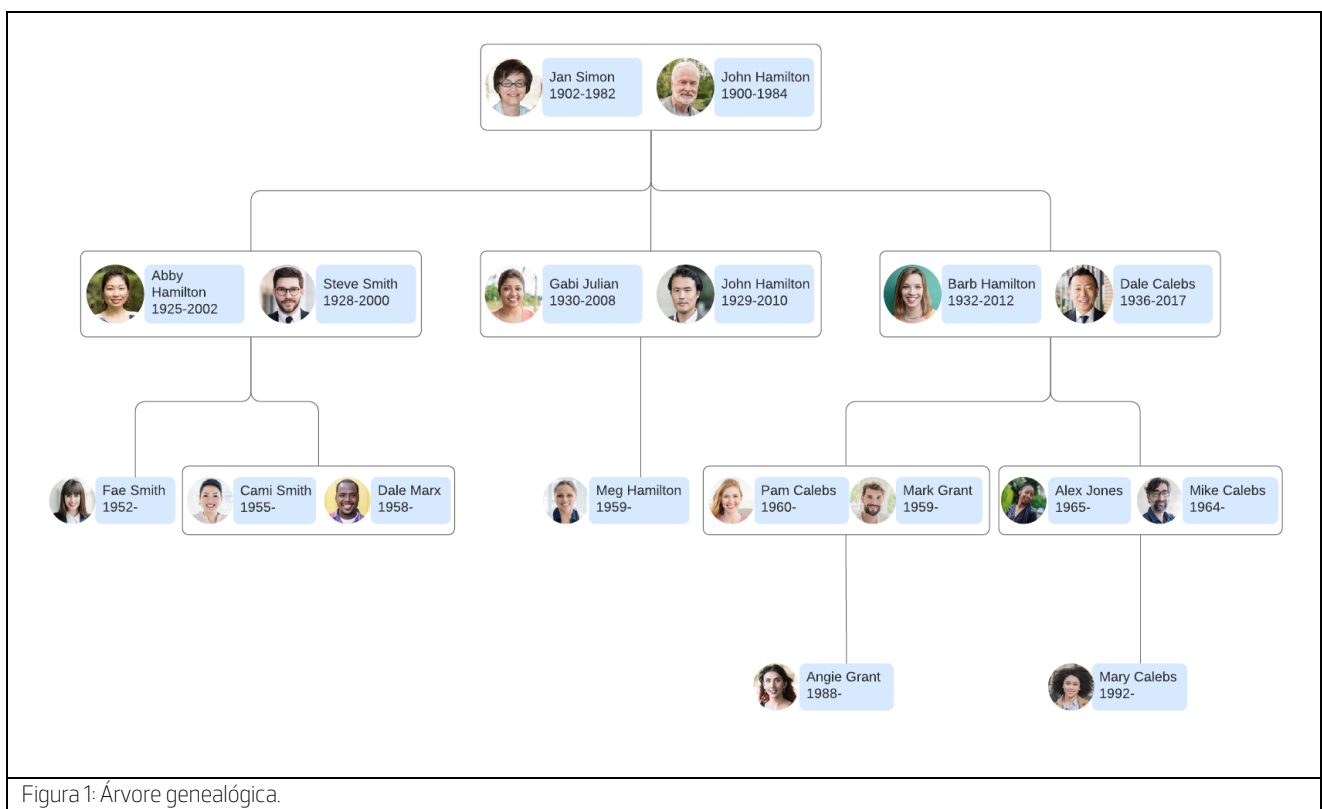


Figura 1: Árvore genealógica.

Exercício 2

Considere as relações entre algumas das personagens da série *Game of Thrones*, tal como ilustrado na Figura 1, ignorando para efeitos deste exercício se as relações aconteceram no passado ou acontecem atualmente.

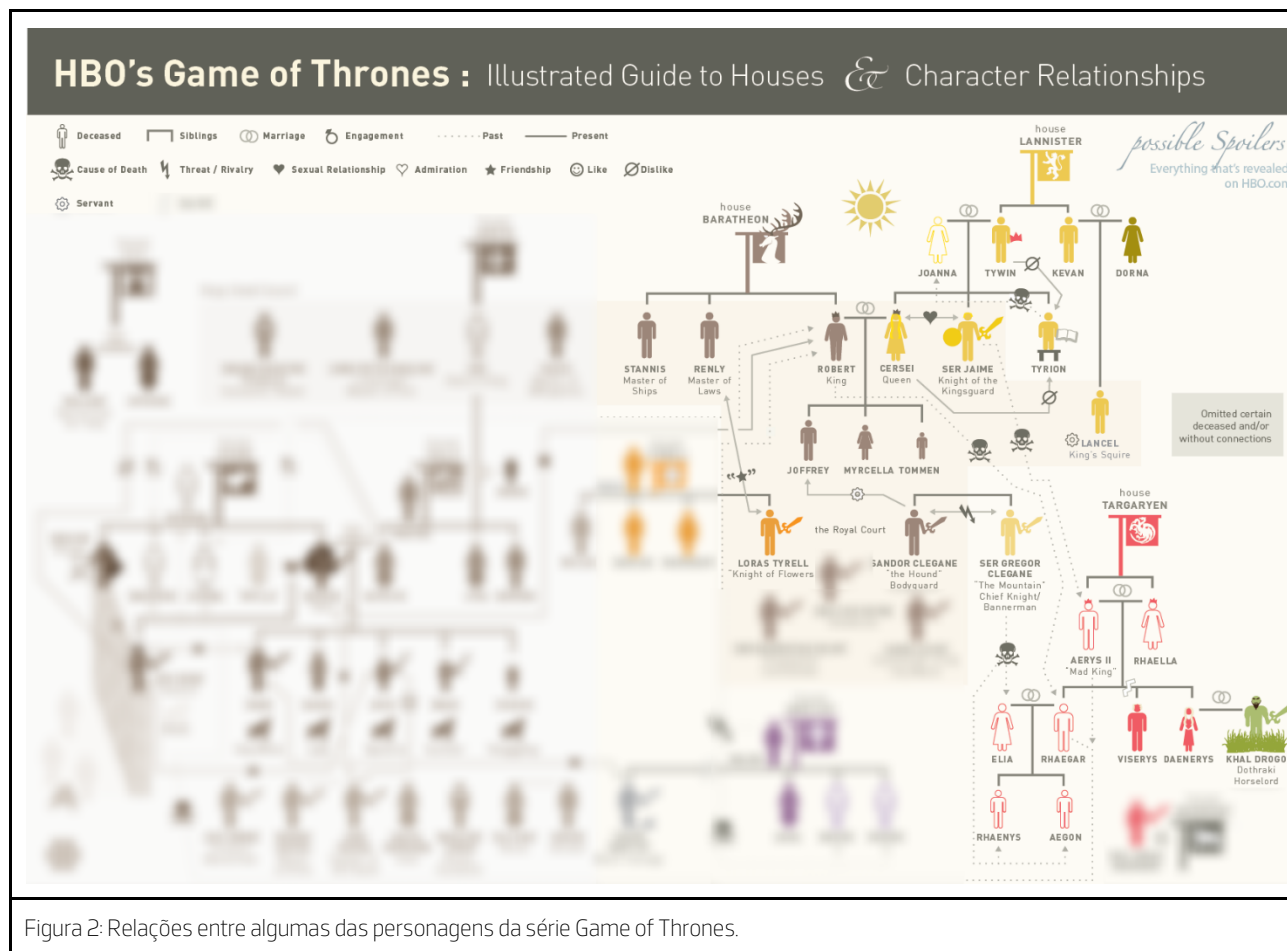



Figura 2: Relações entre algumas das personagens da série Game of Thrones.

Escreva factos/regras Prolog que representem as seguintes relações:


- homem/1
- mulher/1
- família/2 – família(nome, nome_familia). Considere os irmãos Clegane membros da família Baratheon.
- casados/2 – casados(homem, mulher).
- descendenteDir/3 – descendente(homem, mulher, filho).

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>LEI – Licenciatura em Engenharia Informática</p> <p>IA – Inteligência Artificial</p> <p>2º Semestre – Docente: DCarneiro Ficha Prática 1</p>
---	--

- irmao/2 – defina este facto à custa de factos já definidos. Considere que duas personagens apenas têm o relacionamento de irmão se forem ambos filhos do mesmo pai e da mesma mãe.
- matou/2 – matou(assassino, vítima).
- inimigo/2
- amigo/2
- relamorosa/2 – quando não há relação de casamento.
- filho/2 e filha/2 – filho(pai, filho)
- gay/1 – define uma personagem homossexual
- viuvo/1 e viuva/1 – define uma personagem casada com uma personagem já morta
- viuvos_amorosos/2 – duas personagens viúvas, de sexos opostos, que mantêm um relacionamento amoroso.
- vingança_pendente/2 – existe uma vingança pendente entre duas personagens quando a segunda matou o conjugue da primeira
- assassino/1 – determina se uma personagem é assassino
- morto/1 – determina se uma personagem está morta
- vingou/2 – X vingou Y se X matou uma pessoa que tinha previamente morto Y
- descendente/2 – determina se uma personagem é descendente de outra (em qualquer grau)
- infiel/1 – uma personagem é infiel se está casada com uma pessoa (que não esteja morta) mas mantém uma relação amorosa com outra (que também não esteja morta)

Questione, através da consola, esses mesmos factos de forma a responder positivamente ou negativamente às seguintes perguntas:

- Aerys está morto?
- Sir Jaime matou Aerys?
- Tywin é amigo de Tyrion?
- O Stannis é irmão do Renly?
- Existe alguma relação de incesto?
- Liste todos os descendentes de Tywin Lanister.

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>LEI – Licenciatura em Engenharia Informática</p> <p>IA – Inteligência Artificial</p> <p>2º Semestre – Docente: DCarneiro Ficha Prática 1</p>
---	--

Exercício 3

Escreva agora factos/regras que permitam determinar a seguinte informação, através da consola:

- Entre que personagens existem relações de incesto?
- Que personagens foram mortas por Tyrion?
- Que personagens da família Targaryen foram mortas por Ser Jaime?
- Que personagens mataram membros da família Targaryen?
- Assumindo que não sabe a verdade, que personagens são resultado do casamento entre Cersei e Robert?
- Determine todos os filhos de todos os relacionamentos de casamento.

Exercício 4

Considere o contexto do jogo Robocode (Figura 3). Neste jogo, robots virtuais são programados para agirem de forma autónoma, tendo como finalidade encontrar robots inimigos e disparar de forma a retirar-lhes energia. O jogo termina quando apenas um robot sobrevive, sendo esse o vencedor da partida.

Os robots são tipicamente programados em Java. No entanto, é possível modelar-se em Prolog o estado do jogo, e implementar-se um sistema de decisão baseado em regras, que depois é invocado pelo Robocode para tomar decisões em tempo real sobre como o robot se deve movimentar ou disparar.

Neste exercício pretende-se que modele o conhecimento de um momento hipotético do jogo.

Considere a existência dos seguintes factos:

- `posição(X, Y, Inimigo)` – representa a última posição conhecida de um dado inimigo;
- `eu(X,Y)` – representa a posição atual do meu robot;
- `accuracy(Inimigo, P)` – representa a eficácia do robot (em percentagem), contra um determinado inimigo;
- `disparo(Inimigo, Acertou)` – representa um disparo feito a um determinado inimigo e o resultado (hit/no);

Escreva predicados que permitam determinar a seguinte informação:

- `distancia/2` – calcula a distância a que um dado inimigo se encontra do meu robot;
- `distanciaMedia/1` – calcula a distância média a que o meu robot se encontra dos inimigos;
- `totalInimigos/1` – calcula o total de inimigos com os quais estou a combater;
- `maisProximo/1` – determina o nome do inimigo mais próximo;
- `inimigosPerto/2` – determina a lista de inimigos que estão mais próximos que uma dada distância;
- `melhoresInimigos/1` – determina a lista dos inimigos cuja eficácia do meu robot é superior a 75%;

- disparar/3 – determina se o meu robot deve ou não disparar (e com que potência) a um determinado inimigo, que se move a uma dada velocidade, de acordo com as seguintes regras:
 - Sempre que a eficácia contra o inimigo é superior a 75% ou sempre que este se encontra a uma distância inferior a 100 ou sempre que o inimigo é o sitting duck dispara com potência 3;
 - Se a velocidade do inimigo é superior a 4, não dispara (para a frente ou para trás);
 - Se a velocidade do inimigo está entre 2 e 4 (para a frente ou para trás):
 - Se a distância ao inimigo é inferior a 200 dispara com potência 2;
 - Senão dispara com potência 1;
 - Se a velocidade do inimigo está entre 1 e 2 (para a frente ou para trás):
 - Se a distância ao inimigo é inferior a 200 dispara com potência 3;
 - Senão dispara com potência 2;

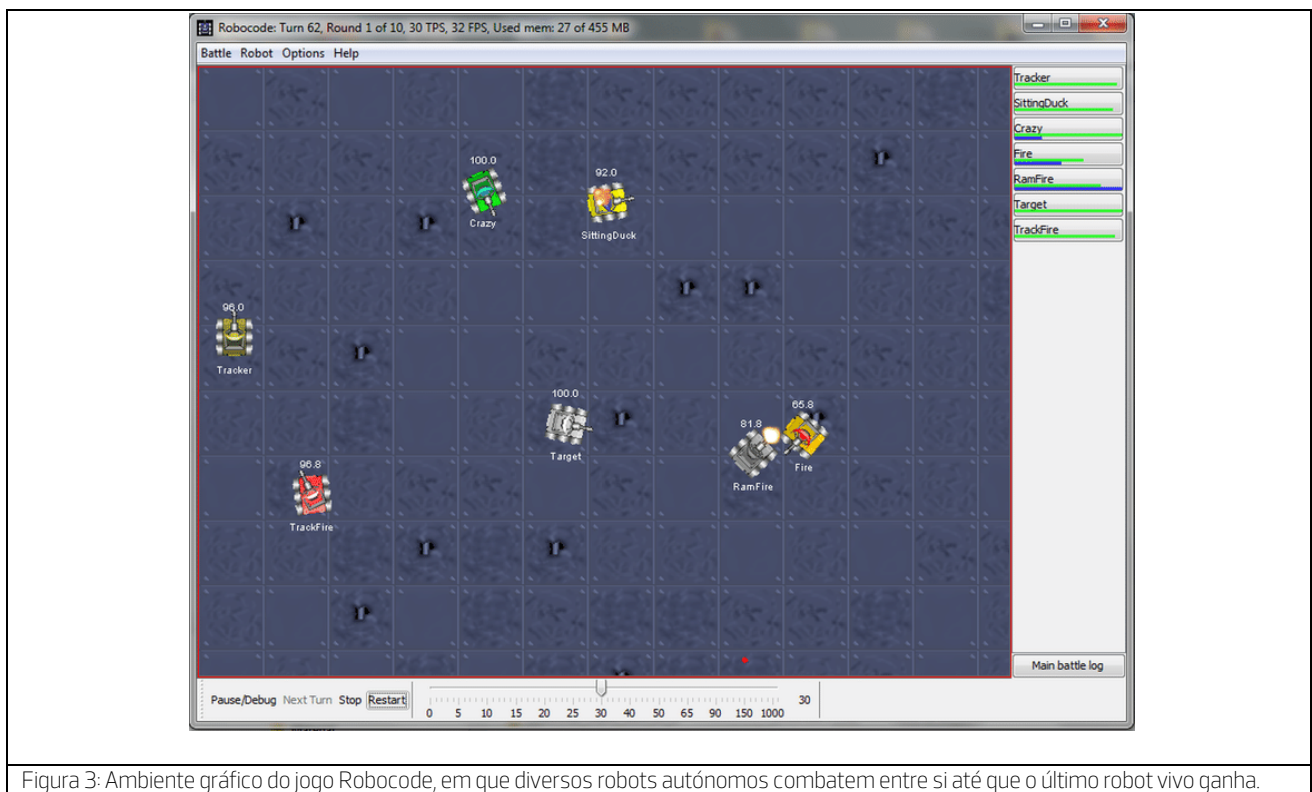


Figura 3: Ambiente gráfico do jogo Robocode, em que diversos robots autónomos combatem entre si até que o último robot vivo ganha.

Exercício 5

Considere a seguinte base de conhecimento em Prolog:

```
aluno(1,joao,m).
aluno(2,antonio,m).
```

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>LEI – Licenciatura em Engenharia Informática</p> <p>IA – Inteligência Artificial</p> <p>2º Semestre – Docente: DCarneiro Ficha Prática 1</p>
---	--

```
aluno(3,carlos,m).
aluno(4,luisa,f).
aluno(5,maria,f).
aluno(6,isabel,f).
```

```
curso(1,lei).
curso(2,lsirc).
curso(3,lsig).
```

```
%disciplina(cod,sigla,ano,curso)
disciplina(1,ed,2,1).
disciplina(2,ia,3,1).
disciplina(3,fp,1,2).
```

```
%inscrito(aluno,disciplina)
inscrito(1,1).
inscrito(1,2).
inscrito(5,3).
inscrito(5,5).
```

```
%nota(aluno,disciplina,nota)
nota(1,1,15).
nota(1,2,16).
nota(1,5,20).
nota(2,5,10).
nota(3,5,8).
```

```
%copia
copia(1,2).
copia(2,3).
copia(3,4).
```

Escreva predicados que permitam saber

- Quais os alunos que não estão inscritos em qualquer disciplina
- Quais os alunos que não estão inscritos em qualquer disciplina, assumindo que um aluno inscrito numa disciplina que não existe não está inscrito
- Qual a média de um determinado aluno
- Quais os alunos cuja média é acima da média (considere todas as notas de todas as disciplinas)
- Quais os nomes dos alunos que copiaram
- Quais os alunos que copiaram (diretamente ou indiretamente) por um dado aluno
- mapToNome - converter uma lista de números de alunos numa lista de nomes. Assuma que podem ser dados números de alunos não registados (que devem ser ignorados)

Exercício 6

Escreva predicados que permitam:

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>LEI – Licenciatura em Engenharia Informática</p> <p>IA – Inteligência Artificial</p> <p>2º Semestre – Docente: DCarneiro Ficha Prática 1</p>
---	--

- Somar os primeiros n números naturais
- Obter o último elemento de uma lista
- Obter o penúltimo elemento de uma lista
- Obter o elemento na posição n de uma lista
- Calcular tamanho de uma lista
- Determinar se uma variável é uma lista
- Inserir um elemento no fim da lista
- Concatenar duas listas
- Eliminar os elementos consecutivos duplicados de uma lista (eliminadupcons([1,2,3,3,4,5,5,6,7],X). resulta em $X = [1, 2, 4, 6, 7]$;))
- Duplicar cada elemento de uma lista
- Retirar o elemento na posição n de uma lista
- Determinar o somatório dos elementos positivos de uma lista de inteiros
- Eliminar a primeira ocorrência de um determinado número de uma lista
- Eliminar todas as ocorrências de um determinado número de uma lista
- Substituir todas as ocorrências de um determinado número numa lista, por outro
- Inserir um número numa determinada posição de uma lista
- Determinar todos os elementos de uma lista que também são elementos de uma outra lista (união)