

## Inteligência Artificial 2014-2015

Teste de SBC. Duração: 1:30H

2014/11/21

*Nota: Nos exercícios que se seguem, a programação em lógica, as regras condição-conclusão, e as regras de produção estão expressas e funcionam tal como nos sistemas usados e explicados nas aulas. As respostas devem seguir os mesmos padrões.*

### I – Representação computacional de conhecimento

Considera os seguintes predicados

ingrediente_disponivel /1	ingrediente_disponivel(Ingrediente) significa que o <i>Ingrediente</i> especificado está disponível
ingrediente_necessario /2	ingrediente_necessario(Cozinhado, Ingrediente) significa que, para confeccionar o <i>Cozinhado</i> especificado, é necessário o <i>Ingrediente</i> especificado
ingrediente_em_falta/2	ingrediente_em_falta(Cozinhado, Ingrediente) significa que, para confeccionar o <i>Cozinhado</i> especificado, falta-te o <i>Ingrediente</i> especificado
aval_cozinhado/2	aval_cozinhado(Cozinhado, Aval) significa que a avaliação do <i>Cozinhado</i> especificado é <i>Aval</i>
melhor_cozinhado/1	melhor_cozinhado(Cozinhado) significa que <i>Cozinhado</i> é o melhor cozinhado, em termos de avaliação
vende/2	vende(Loja, Ingrediente) significa que o Ingrediente especificado existe à venda na Loja especificada
ingrediente_loja/2	ingrediente_loja(Ingrediente, Loja) significa que o Ingrediente especificado deve ser comprado na Loja especificada. Estes factos servem de input à ação comprar/1 do robot Fredo.
visitada/1	visitada(Loja) significa que a Loja especificada já foi visitada

(2.5 Valores) 1 - Escreve a parte da base de conhecimento *BRules* correspondente á seguinte informação. Os teus ingredientes disponíveis são leite de coco, cominhos e azeite. Para confeccionar caril, tens de ter leite de coco, cominhos e magia. Para confeccionar bitoque, tens de ter bife e ovo.

#### R:

```
fact(ingrediente_disponivel(leite_de_coco)).
fact(ingrediente_disponivel(cominhos)).
fact(ingrediente_disponivel(azeite)).

fact(ingrediente_necessario(caril, leite_de_coco)).
fact(ingrediente_necessario(caril, cominhos)).
```

```
fact(ingrediente_necessario(caril, magia)).

fact(ingrediente_necessario(bitoque, bife)).

fact(ingrediente_necessario(bitoque, ovo)).
```

(2.5 Valores) 2 - Admitindo que tens informação sobre os ingredientes disponíveis para cozinhar, e os ingredientes requeridos para cada cozinhado, define uma base de conhecimento, recorrendo à representação por regras condição-conclusão, capaz de determinar que ingredientes faltam para um determinado cozinhado, por exemplo

```
?- solve(ingrediente_em_falta(Coz, Ingr)).
Coz = caril      Ingr = magia ;
Coz = bitoque   Ingr = bife ;
Coz = bitoque   Ingr = ovo ;
false
```

**R:**

```
if    (ingrediente_necessario(Coz, Ingrediente) and
      not(ingrediente_disponivel(Ingrediente)))
then ingrediente_em_falta(Coz, Ingrediente).
```

(2.5 Valores) 3 - Representa exatamente o mesmo conhecimento da pergunta 2, mas em Prolog, de modo que a interrogação exemplificada na pergunta 2 fosse escrita da seguinte forma:

```
?- ingrediente_em_falta(Coz, Ingr).
Coz = caril      Ingr = magia ;
Coz = bitoque   Ingr = bife ;
Coz = bitoque   Ingr = ovo ;
false
```

**R:**

```
ingrediente_em_falta(Coz, Ingrediente):-
    ingrediente_necessario(Coz, Ingrediente),
    \+ ingrediente_disponivel(Ingrediente).
```

(2.5 Valores) 4 - Admitindo que tens informação sobre a avaliação de vários cozinhados (*aval\_cozinhado/2*), define uma base de conhecimento, recorrendo à representação por regras condição-conclusão, capaz de determinar o melhor cozinhado que podes fazer, por exemplo

```
?- solve(melhor_cozinhado(Cozinhado)).
Cozinhado = pizza
```

**R:**

```
if    (aval_cozinhado(Coz, Aval) and
      not( (aval_cozinhado(_, A) and A > Aval) ))
then melhor_cozinhado(Coz).
```

(2.5 Valores) 5 - Estás a construir o *robot* Fredo para fazer compras e necessitas escrever um sistema de regras de produção para controlar o seu comportamento.

Escreve a ou as regras de uma base de conhecimentos para mandar o Fredo comprar os ingredientes em falta para o melhor cozinhado. Sugestão: criar primeiro os factos *ingrediente\_loja/2* e, depois, mandar o *robot* comprar.

fredo\_compra é o programa de interface com o sistema baseado em conhecimento.

```
fredo_compra :-
    psys,
    retractall(visitada(_)).
```

Além dos predicados já especificados, o sistema de controlo do robot Fredo pode usar as seguintes ações já implementadas.

assert/1	assert(Facto) adiciona o <i>Facto</i> especificado à memória
retractall/1	retractall(Facto) remove da memória todos os factos que emparelhem com o <i>Facto</i> especificado. Nunca falha, mesmo que nenhum facto emparelhe com o especificado.
comprar/1	comprar(Loja) compra, na <i>Loja</i> especificada, todos os ingredientes especificados em factos ingrediente_loja(Ingrediente, Loja). Regista o facto visitada(Loja).

**R:**

```
if    (melhor_cozinhado(Coz) and
       ingrediente_em_falta(Coz, Ingr) and
       vende(Loja, Ingr) and
       \+ ingrediente_loja(Ingr, Loja))
then assert(ingrediente_loja(Ingr, Loja)).

if    (ingrediente_loja(_, Loja) and \+ visitada(Loja))
then (comprar(Loja), retractall(ingrediente_loja(_, Loja))).
```

## II – Lógica de Predicados

- 1)  $\forall x [(Mulher(x) \wedge \exists y Filho(y, x)) \Rightarrow Mae(x)]$   $\Delta$
- 2)  $Mulher(Ana)$   $\Delta$
- 3)  $Filho(Rui, Ana)$   $\Delta$

(2.5 Valores) 6 - Considera a base de conhecimentos apresentada. Mostra que se deriva dessa base de conhecimento que a Ana é uma mãe.

**R:**

- 1)  $\forall x [(Mulher(x) \wedge \exists y Filho(y, x)) \Rightarrow Mae(x)]$   $\Delta$
- 2)  $Mulher(Ana)$   $\Delta$
- 3)  $Filho(Rui, Ana)$   $\Delta$
- 4)  $\exists y Filho(y, Ana)$  3 EG
- 5)  $(Mulher(Ana) \wedge \exists y Filho(y, Ana)) \Rightarrow Mae(Ana)$  1 UI
- 6)  $Mulher(Ana) \wedge \exists y Filho(y, Ana)$  2, 4 AI
- 7)  $Mae(Ana)$  5, 6 MP

(2.5 Valores) 7 - Usando prova por refutação na forma clausal, mostra que a Ana é mãe. Para isso, assume que

$Mulher(Ana)$  equivale à cláusula  $\{ Mulher(Ana) \}$

$Filho(Rui, Ana)$  equivale à cláusula  $\{ Filho(Rui, Ana) \}$

$\neg Mae(Ana)$  equivale à cláusula  $\{ \neg Mae(Ana) \}$

## **R:**

Para provar por refutação na forma clausal, é necessário (i) converter a BC em forma clausal; (ii) negar o objetivo e convertê-lo em forma clausal; e (iii) mostrar que o conjunto de cláusulas obtido é contraditório.

### Converter a BC em forma clausal

Conhecendo as cláusulas correspondentes a Mulher(Ana) e a Filho(Rui, Ana), basta converter a primeira proposição.

$$\forall x [(Mulher(x) \wedge \exists y Filho(y, x)) \Rightarrow Mae(x)]$$

Substituir  $(A \Rightarrow B)$  por  $(\neg A \vee B)$

$$\forall x [\neg(Mulher(x) \wedge \exists y Filho(y, x)) \vee Mae(x)]$$

Mover as negações para os literais  $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$ ;  $(\neg \exists x A) \equiv (\forall x \neg A)$

$$\forall x [(\neg Mulher(x) \vee \neg \exists y Filho(y, x)) \vee Mae(x)]$$

$$\forall x [(\neg Mulher(x) \vee \forall y \neg Filho(y, x)) \vee Mae(x)]$$

Skolemização: N/A porque não existem quantificadores existenciais

Remover os  $\forall$ s

$$\neg Mulher(x) \vee \neg Filho(y, x) \vee Mae(x)$$

Escrever em forma de uma conjunção de disjunções. N/A porque já temos uma única disjunção

Escrever em forma de cláusulas

$$\{\neg Mulher(x), \neg Filho(y, x), Mae(x)\}$$

### Negar o objetivo e converter em forma clausal

Objetivo: Mae(Ana)

Objetivo negado:  $\neg Mae(Ana)$

Objetivo em forma clausal:  $\{\neg Mae(Ana)\}$  [é um dado do enunciado]

Mostrar que se tem um conjunto de cláusulas contraditório

1)	$\{\neg Mulher(x), \neg Filho(y, x), Mae(x)\}$	$\Delta$
2)	$\{Mulher(Ana)\}$	$\Delta$
3)	$\{Filho(Rui, Ana)\}$	$\Delta$
4)	$\{\neg Mae(Ana)\}$	$\neg$ Objetivo
5)	$\{\neg Filho(y, Ana), Mae(Ana)\}$	1, 2
6)	$\{Mae(Ana)\}$	5, 3
7)	$\{\}$	6, 4

## **III – E ainda...**

8 – Responde, sem justificar, às seguintes perguntas, sem justificar. Nos casos assinalados, as respostas erradas descontam de acordo com o indicado.

(1 Valor) a) Que tipo de raciocínio se aplica no sistema da pergunta 5, para a frente ou para trás? **[desconta 0.5]**

(1 Valor) b) Uma regra do domínio de um sistema baseado em regras deve fazer parte da base de conhecimento ou do motor de inferência? **[desconta 0.5]**

(0.5 Valores) c) Como se chama o conjunto de regras com condição satisfeita e que podem ser aplicadas num determinado momento? [**não desconta**]

**R:**

- a) Para a frente
- b) Na base de conhecimento
- c) Conjunto de Conflito