

Inteligência Artificial 2017-2018

Teste de SBC

2017/11/25, 01:30H

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

Base de conhecimento e conceitos envolvidos

Considera a base de conhecimentos que se segue e os conceitos nela envolvidos, os quais servirão de base para as perguntas do teste.

1. Catapultado(Luis, 10)
2. Catapultado(Isabel, 9)
3. Catapultado(XI27, 7)
4. Professor(Luis)
5. Professor(Isabel)
6. GalináceoMecânico(XI27)
7. $\forall_x \forall_t [(Catapultado(x, t) \vee Esmagado(x, t)) \Rightarrow Assassinado(x, t)]$

Catapultado(X, T)	X foi catapultado às T horas
Professor(X)	X é um professor
GalináceoMecânico(X)	X é um galináceo mecânico
Assassinado(X, T)	X foi assassinado às T horas
Esmagado(X, T)	X foi esmagado às T horas

I – Lógica de Predicados

(3.5 Valores) 1 – Sem recorrer à forma clausal, mostra que, a partir da base de conhecimento especificada, se deduz que existe pelo menos um instante de tempo em que pelo menos um professor foi assassinado: $\exists_t \exists_p (Professor(p) \wedge Assassinado(p, t))$.

R:

Objetivo: $\exists_t \exists_p (Professor(p) \wedge Assassinado(p, t))$

8. $(Catapultado(Luis, 10) \vee Esmagado(Luis, 10))$ 1 OI
9. $[(Catapultado(Luis, 10) \vee Esmagado(Luis, 10)) \Rightarrow Assassinado(Luis, 10)]$
7 UI $\times 2$
10. $Assassinado(Luis, 10)$ 9, 8 MP
11. $(Professor(Luis) \wedge Assassinado(Luis, 10))$ 4, 10 AI
12. $\exists_t \exists_p (Professor(p) \wedge Assassinado(p, t))$ 11 EG $\times 2$

(3.5 Valores) 2 – Representa o seguinte conhecimento na linguagem de Lógica de Predicados de Primeira Ordem:

Para todo e qualquer professor, existe um instante de tempo em que ele é catapultado.

R:

$$\forall_x[\exists_t(Professor(x) \Rightarrow Catapultado(x, t))]$$

ou

$$\forall_x[Professor(x) \Rightarrow \exists_t Catapultado(x, t)]$$

II – Programação em lógica

Considera os conceitos apresentados e exemplificados no início do enunciado e responde às seguintes perguntas. Repara que tens de adaptar os símbolos usados para respeitar as convenções sintáticas do Prolog: os nomes das variáveis começam com maiúsculas em vez de minúsculas, e os nomes das constantes começam com minúsculas.

(3.5 Valores) 3 – Define o predicado *prof_assassinado/2*, tal que *prof_assassinado(X, T)* significa que o professor X foi assassinado à hora T. Para tal, também tens de representar, em Prolog, a proposição 7 da base de conhecimentos

R:

```
prof_assassinado(X, T) :-  
    professor(X),  
    assassinado(X, T).
```

```
assassinado(X, T) :-  
    catapultado(X, T).
```

```
assassinado(X, T) :-  
    esmagado(X, T).
```

(3 Valores) 4 – Admitindo que o predicado *prof_assassinado/2* da pergunta anterior já está corretamente implementado, define o predicado *primeiro_prof_assassinado/1*, tal que *primeiro_prof_assassinado(X)* significa que X foi o professor assassinado mais cedo.

R:

```
primeiro_prof_assassinado(X) :-  
    prof_assassinado(X, T),  
    \+ existe_prof_assassinado_ha_mais_tempo(T).
```

```
existe_prof_assassinado_ha_mais_tempo(T) :-  
    prof_assassinado(_, Q),  
    T > Q.
```

Ou

```
primeiro_prof_assassinado(X) :-  
    prof_assassinado(X, T),  
    \+ ( prof_assassinado(_, Q), T > Q ).
```

III – Regras de produção

Tendo em conta as ações da tabela e os conceitos apresentados e exemplificados no início do enunciado, responde às perguntas que se seguem. As tuas soluções não podem destruir os factos previamente existentes na base de conhecimento.

Repara que tens de adaptar os símbolos usados para respeitar as convenções sintáticas do PSys: os nomes das variáveis começam com maiúsculas em vez de minúsculas, e os nomes das constantes começam com minúsculas.

catapultar(X)	Catapulta X. Cria o facto catapultado(X, T) com a identidade da vítima e a hora de execução.
esmagar(X)	Esmaga X. Cria o facto esmagado(X, T).
assert(Clausula)	Cria a cláusula especificada, a qual é acrescentada à memória
retract(Clausula)	Remove, da memória, a primeira cláusula que emparelha com a cláusula especificada, ou falha se nenhuma emparelhar

(3.5 Valores) 5 – Escreve as regras de produção que catapultam todos os professores e esmagam todos os galináceos mecânicos, sem nenhuma restrição de ordem.

R:

```
if (professor(X) and \+ catapultado(X, _))
then catapultar(X) .

if (galinaceo_mecanico(X) and \+ esmagado(X, _))
then esmagar(X) .
```

(3 Valores) 6 – Escreve as regras de produção que, primeiro catapultam todos os professores, e depois esborracham todos os galináceos mecânicos. As regras que escreveres terão de criar o facto de controlo *todos_os_profs_assassinados*, quando todos os professores tiverem sido catapultados. É importante que tenhas presente que um conjunto de regras estará correto apenas se o seu comportamento for o desejado, mesmo que a ordem pela qual as regras são escritas seja alterada.

R:

```
if (professor(X) and \+ catapultado(X, _))
then catapultar(X) .

if (\+ todos_os_profs_assassinados and
    \+ ( professor(X) and \+ catapultado(X, _)) )
then assert(todos_os_profs_assassinados) .

if (todos_os_profs_assassinados and
    galinaceo_mecanico(Y) and \+ esmagado(Y, _))
then esmagar(Y) .
```