

Lógica para Programação

Segundo Teste

17 de Junho de 2009

13:00–14:30

Nome: _____ Número: _____

- Esta prova, individual e sem consulta, tem 7 páginas com 13 perguntas. A cotação de cada pergunta está assinalada entre parêntesis.
- Escreva o seu número em todas as folhas da prova. O tamanho das respostas deve ser limitado ao espaço fornecido para cada questão. O corpo docente reserva-se o direito de não considerar a parte das respostas que excedam o espaço indicado.
- Pode responder usando lápis.
- Em cima da mesa devem apenas estar o enunciado, caneta ou lápis e borracha e cartão de aluno. Não é permitida a utilização de folhas de rascunho, telemóveis, calculadoras, etc.
- Boa sorte.

Pergunta	Cotação	Nota
1.	1.0	
2.	1.0	
3.	1.0	
4.	1.0	
5.	1.0	
6.	1.5	
7.	1.0	
8.	1.5	
9.	1.0	
10.	3.0	
11.	1.0	
12.	3.0	
13.	3.0	
Total	20.0	

1. (1.0) Dada uma conceptualização (D, F, R) , diga quais as condições a que uma interpretação, I , deve obedecer.

Resposta:

- (a) Cada constante individual f_i^0 é associada com uma entidade do universo de discurso D ;
- (b) Cada letra de função f_i^n é associada a uma função de F . Se f_i^n é uma letra de função com aridade n , correspondendo à função $I(f_i^n)$ da conceptualização, e se t_1, \dots, t_n são termos, então $f_i^n(t_1, \dots, t_n)$ corresponde à entidade $I(f_i^n)(I(t_1), \dots, I(t_n))$ da conceptualização;
- (c) A cada letra de predicado P_i^n é associada uma relação de R .

2. No contexto da programação em lógica, sendo Δ um programa e α um objectivo, diga o que é:

- (a) (0.5) Uma resposta de Δ ao objectivo α .

Resposta:

É uma substituição s para as variáveis de α .

- (b) (0.5) Uma resposta correcta de Δ ao objectivo α .

Resposta:

É uma resposta s de Δ ao objectivo α tal que $\Delta \models (\alpha \cdot s)$.

3. O PROLOG utiliza a resolução SLD com uma função específica de selecção e com uma regra específica de procura.

- (a) (0.5) Qual a função de selecção utilizada em PROLOG?

Resposta:

A função de selecção que escolhe o primeiro literal na cláusula objectivo.

- (b) (0.5) Qual a regra de procura utilizada em PROLOG?

Resposta:

A regra de procura que escolhe a primeira cláusula unificável com o literal seleccionado da cláusula objectivo na sequência de cláusulas que corresponde ao programa.

4. (1.0) Diga o que se entende em PROLOG por uma cláusula iterativa.

Resposta:

Uma cláusula iterativa é uma cláusula cujo corpo apenas contém um literal, usando o mesmo predicado que o utilizado na cabeça da cláusula. Antes desse literal podem existir zero ou mais utilizações de predicados pré-definidos.

5. (1.0) Considere a conceptualização:

- *Universo de discurso.* $D = \{\spadesuit, \heartsuit\}$
- *Conjunto de funções.* $F = \{(\spadesuit, \heartsuit), (\heartsuit, \spadesuit)\}$
- *Conjunto de relações.* $R = \{(\spadesuit, \spadesuit), (\spadesuit, \heartsuit)\}$

e a seguinte interpretação:

$I(a) \mapsto \spadesuit$

$I(b) \mapsto \heartsuit$

$I(f) \mapsto \{(\spadesuit, \heartsuit), (\heartsuit, \spadesuit)\}$

$I(P) \mapsto \{(\spadesuit, \spadesuit), (\spadesuit, \heartsuit)\}$

Diga, justificando, se a seguinte *fbf* é satisfeita pela interpretação I .

$$P(a, f(a))$$

Resposta:

A interpretação I satisfaz a *fbf* $P(a, f(a))$, se e só se $(I(a), I(f(a)))$ for um elemento da relação $I(P)$. Como $(I(a), I(f(a))) = (\spadesuit, I(f)(\spadesuit)) = (\spadesuit, \heartsuit)$ e como $(\spadesuit, \heartsuit) \in \{(\spadesuit, \spadesuit), (\spadesuit, \heartsuit)\}$, a interpretação satisfaz a *fbf* $P(a, f(a))$.

6. (1.5) Considere o seguinte conjunto de cláusulas de Horn:

$$P(x, y) \leftarrow Q(x, y)$$

$$P(x, y) \leftarrow R(x, y), S(x)$$

$$R(x, y) \leftarrow S(x), T(y)$$

$$P(a, b) \leftarrow$$

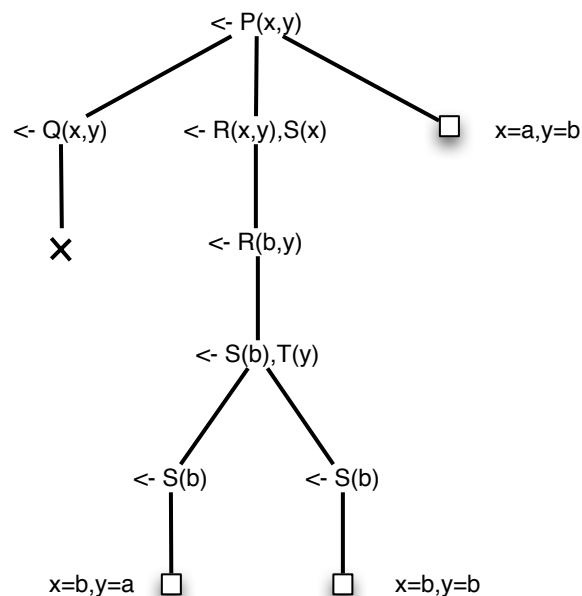
$$S(b) \leftarrow$$

$$T(a) \leftarrow$$

$$T(b) \leftarrow$$

Usando uma árvore de resolução SLD e uma função de selecção que escolha para unificar o *último* literal do objectivo, mostre todas as soluções para o seguinte objectivo: $\leftarrow P(x, y)$. Pode usar a estratégia de procura que preferir. No final indique explicitamente todas as soluções.

Resposta:



Este objectivo tem três soluções: $x = b$ e $y = a$, $x = b$ e $y = b$, $x = a$ e $y = b$.

7. (1.0) Utilize o algoritmo de unificação para determinar se o seguinte conjunto de *fbfs* é unificável, e, no caso de o ser, determine o unificador mais geral. Mostre todos os passos intermédios usados nos cálculos.

$$\{D(x_1, b, p(n)), D(j, x_2, p(y_2)), D(x_3, b, y_3)\}.$$

Resposta:

Conjunto de <i>fbfs</i>	Conj. desacordo	Substituição
$\{D(x_1, b, p(n)), D(j, x_2, p(y_2)), D(x_3, b, y_3)\}$	$\{x_1, j, x_3\}$	$\{j/x_1\}$
$\{D(j, b, p(n)), D(j, x_2, p(y_2)), D(x_3, b, y_3)\}$	$\{j, x_3\}$	$\{j/x_3\}$
$\{D(j, b, p(n)), D(j, x_2, p(y_2)), D(j, b, y_3)\}$	$\{b, x_2\}$	$\{b/x_2\}$
$\{D(j, b, p(n)), D(j, b, p(y_2)), D(j, b, y_3)\}$	$\{p(n), p(y_2), y_3\}$	$\{p(n)/y_3\}$
$\{D(j, b, p(n)), D(j, b, p(y_2))\}$	$\{n, y_2\}$	$\{n/y_2\}$
$\{D(j, b, p(n))\}$		

O unificador mais geral é $\{n/y_2, p(n)/y_3, b/x_2, j/x_1, j/x_3\}$.

8. (1.5) Transforme a seguinte *fbf* da lógica de primeira ordem em forma clausal.

$$\exists x[A(x)] \wedge \forall x, y[B(x) \rightarrow (\exists w[C(y, x, w)] \wedge \exists z[D(z, x)])] \wedge \exists x[E(x) \vee F(x)]$$

Resposta:(a) Eliminação de \rightarrow :

$$\exists x[A(x)] \wedge \forall x, y[\neg B(x) \vee (\exists w[C(y, x, w)] \wedge \exists z[D(z, x)])] \wedge \exists x[E(x) \vee F(x)]$$

(b) Redução do domínio de \neg : já está.

(c) Normalização de variáveis:

$$\exists x[A(x)] \wedge \forall z, y[\neg B(z) \vee \exists w[C(y, z, w)] \wedge \exists r[D(r, z)]] \wedge (\exists s[E(s) \vee F(s)])$$

(d) Eliminação de \exists :

$$A(sk_1) \wedge \forall z, y[\neg B(z) \vee (C(y, z, skf_1(z, y)) \wedge D(skf_2(z, y), z))] \wedge (E(sk_2) \vee F(sk_2))$$

(e) Mover \forall para esquerda:

$$\forall z, y[A(sk_1) \wedge (\neg B(z) \vee (C(y, z, skf_1(z, y)) \wedge D(skf_2(z, y), z))) \wedge (E(sk_2) \vee F(sk_2))]$$

(f) Eliminação de \forall :

$$A(sk_1) \wedge (\neg B(z) \vee (C(y, z, skf_1(z, y)) \wedge D(skf_2(z, y), z))) \wedge (E(sk_2) \vee F(sk_2))$$

(g) Obtenção da forma conjuntiva normal:

$$A(sk_1) \wedge (\neg B(z) \vee C(y, z, skf_1(z, y))) \wedge (\neg B(z) \vee D(skf_2(z, y), z)) \wedge (E(sk_2) \vee F(sk_2))$$

(h) Eliminação de \wedge :

$$\{A(sk_1), \neg B(z) \vee C(y, z, skf_1(z, y)), \neg B(z) \vee D(skf_2(z, y), z), (E(sk_2) \vee F(sk_2))\}$$

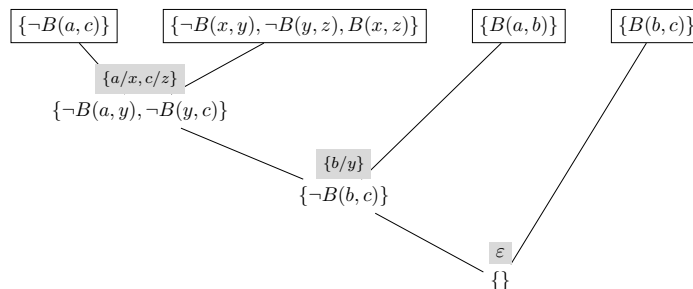
(i) Eliminação de \vee :

$$\{A(sk_1)\}, \{\neg B(z), C(y, z, skf_1(z, y))\}, \{\neg B(z), D(skf_2(z, y), z)\}, \{E(sk_2), F(sk_2)\}$$

9. (1.0) Considere o seguinte conjunto de cláusulas:

$$\{\{\neg B(x, y), \neg B(y, z), B(x, z)\}, \{B(a, b)\}, \{B(b, c)\}\}.$$

Apresente uma demonstração por refutação para $\{B(a, c)\}$ a partir desse conjunto.

Resposta:

10. Considere o seguinte programa em PROLOG (no qual c_1, \dots, c_{10} são identificadores de cláusulas e não pertencem ao programa):

```
c1:  assisteAula1(X, Y) :- aluno(X), disciplina(Y).
c2:  assisteAula2(X, Y) :- aluno(X), !, disciplina(Y).
c3:  assisteAula3(X, Y) :- aluno(X), disciplina(Y), not(odeia(X, Y)).
c4:  assisteAula4(X, Y) :- aluno(X), \+(odeia(X, Y)), disciplina(Y).
c5:  assisteAula5(X, Y) :- \+(odeia(X, Y)), aluno(X), disciplina(Y).
c6:  aluno(maria).
c7:  aluno(afonso).
c8:  disciplina(lp).
c9:  disciplina(ss).
c10: odeia(francisca, lp).
```

- (a) (0.5) Quais das cláusulas anteriores c_1, \dots, c_{10} são:

i. factos

Resposta:

$c_6, c_7, c_8, c_9, c_{10}$

ii. regras

Resposta:

c_1, c_2, c_3, c_4, c_5

- (b) (1.5) Indique *todas* as respostas do PROLOG aos seguintes objectivos (assuma que o utilizador vai escrever ; até esgotar todas as respostas):

i. $?- \text{aluno}(X).$

Resposta:

$X = \text{maria} ;$

$X = \text{afonso}$

ii. $?- \text{assisteAula1}(X, Y).$

Resposta:

$X = \text{maria}, Y = \text{lp} ;$

$X = \text{maria}, Y = \text{ss} ;$

$X = \text{afonso}, Y = \text{lp} ;$

$X = \text{afonso}, Y = \text{ss}$

iii. $?- \text{assisteAula2}(X, Y).$

Resposta:

$X = \text{maria}, Y = \text{lp} ;$

$X = \text{maria}, Y = \text{ss}$

iv. $?- \text{assisteAula3}(X, Y).$

Resposta:

$X = \text{maria}, Y = \text{lp} ;$

$X = \text{maria}, Y = \text{ss} ;$

$X = \text{afonso}, Y = \text{lp} ;$

$X = \text{afonso}, Y = \text{ss}$

v. $?- \text{assisteAula4}(X, Y).$

Resposta:

$X = \text{maria}, Y = \text{lp} ;$

$X = \text{maria}, Y = \text{ss} ;$

$X = \text{afonso}, Y = \text{lp} ;$

$X = \text{afonso}, Y = \text{ss}$

vi. $?- \text{assisteAula5}(X, Y).$

Resposta:

No

- (c) (1.0) Justifique as respostas obtidas nas alíneas (iii) e (v).

Resposta:

Relativamente à alínea (iii), o corte leva a que as alternativas a `aluno(X)` (nomeadamente `aluno(afonso)`) não sejam exploradas, pelo que a única resposta é quando `X = maria` (os ramos abaixo, relativos à unificação do predicado `disciplina/1`, mantêm-se inalterados).

Quanto à alínea (v), objectivo `aluno(X)` unifica com `aluno(maria)`. O facto da variável `Y` não estar instanciada leva a que o objectivo `\+(odeia(X, Y))` tenha sucesso (dado que a `maria` não odeia nenhuma disciplina na base de conhecimento). O objectivo `disciplina(Y)` unifica com `disciplina(lp)`. Após o retrocesso, o objectivo `disciplina(Y)` unifica com `disciplina(ss)`.

Um novo retrocesso origina que objectivo `aluno(X)` unifique com `aluno(afonso)` e o processo anterior repete-se.

11. (1.0) Considere definido o predicado `junta(L1, L2, L3)`, em que `L3` é o resultado de concatenar as listas `L1` e `L2`. Indique, justificando, a que operação sobre listas corresponde o seguinte predicado:

```
xpto(X, L) :- junta(_, [X], L)
```

Resposta:

O predicado `xpto(X, L)` afirma que `X` é o último elemento da lista `L`.

12. Implemente em PROLOG:

- (a) (1.5) O predicado `separa(Lista_or, Lista_pos, Lista_neg)`, aplicável a listas contendo valores numéricos, e que afirma que as listas `Lista_pos` e `Lista_neg` contêm, respectivamente, os elementos positivos e negativos da lista `Lista_or`. Considere que zero é um número positivo.

Resposta:

```
separa([], [], []).
separa([X | Lista_or], [X | Lista_pos], Lista_neg) :-
    X >= 0,
    separa(Lista_or, Lista_pos, Lista_neg).
separa([X | Lista_or], Lista_pos, [X | Lista_neg]) :-
    X < 0,
    separa(Lista_or, Lista_pos, Lista_neg).
```

- (b) (1.5) O predicado `f` que implementa a função $f(n)$ definida recursivamente do seguinte modo:

$$f(n) = \begin{cases} 0 & \text{se } n = 0 \\ f(n-1) + 2n - 1 & \text{se } n > 0 \end{cases}$$

Resposta:

```
f(0, 0).
f(N, Result) :-
    N > 0,
    NAux is N - 1,
    f(NAux, ResultAux),
    Result is ResultAux + 2*N - 1.
```

13. (3.0) Considerando o predicado `liga/6` utilizado no projecto, o qual relaciona informação sobre as ligações directas existentes num mapa, recorde que: (1) este predicado é utilizado com a sintaxe `liga(loc1, loc2, id, dist, vel, port)`; (2) dadas duas localidades quaisquer, existe no máximo uma ligação directa entre essas duas localidades; e (3) qualquer estrada pode ser percorrida nos dois sentidos, estando só um deles indicado no ficheiro que é fornecido ao seu programa.

Defina o predicado `duas_ligacoes/2`. A expressão `duas_ligacoes(loc1, loc2)` afirma que as localidades distintas `loc1` e `loc2` estão ligadas *exactamente* por duas ligações directas, ou seja, é necessário percorrer duas ligações directas para ir da localidade `loc1` à localidade `loc2`.

Resposta:

```
duas_ligacoes(X, Y) :-  
    it_elem(X, Z, _, _, _, _),  
    it_elem(Z, Y, _, _, _, _),  
    X \== Y.  
  
it_elem(X, Y, _, _, _, _) :- liga(X, Y, _, _, _, _).  
it_elem(X, Y, _, _, _, _) :- liga(Y, X, _, _, _, _).
```