



**TÉCNICO**  
LISBOA

# Lógica para Programação

Exame de 1ª Época

23 de Junho de 2021

9:00–11:00

Nome: \_\_\_\_\_ Número: \_\_\_\_\_

- Esta prova, individual e sem consulta, tem **12** páginas com **11** 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 utilizando 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.	1.5	
4.	2.0	
5.	2.0	
6.	4.0	
7.	2.0	
8.	1.0	
9.	1.5	
10.	1.5	
11.	2.5	
Total	20.0	

1. (1.0) Para cada uma das seguintes afirmações, diga se é verdadeira (V) ou falsa (F). Cada resposta correcta vale 0.5 valores e *cada resposta errada desconta 0.2 valores*.

(a) Para quaisquer conjunto de *fbfs*  $\Delta$  e *fbf*  $\alpha$ , se  $\Delta \models \alpha$ , então  $\Delta \not\models \neg\alpha$ .

Resposta: \_\_\_\_

**Resposta:**

F

(b) Para quaisquer *fbfs*  $\alpha_1, \dots, \alpha_n, \beta$ , se o conjunto  $\{\alpha_1, \dots, \alpha_n, \beta\}$  for satisfazível, então  $\{\alpha_1, \dots, \alpha_n\} \models \beta$ .

Resposta: \_\_\_\_

**Resposta:**

F

2. (1.0) Para cada uma das seguintes afirmações, indique a única resposta certa. Cada resposta correcta vale 0.5 valores e *cada resposta errada desconta 0.1 valores*. Considere os seguintes predicados/funções/constantes:

*serieFavorita*( $x$ ) é uma função que devolve a série favorita de  $x$

*Estudante*( $x$ ) é um predicado que é verdade se  $x$  for um estudante

*GostaDe*( $x, y$ ) é um predicado que é verdade se  $x$  gosta de  $y$

*Ana* é uma constante

Indique a fórmula em Lógica de Primeira Ordem que melhor traduz as seguintes frases em Língua Natural:

(a) *Todos os estudantes gostam da Ana*

A:  $\forall x[GostaDe(x, Ana) \rightarrow Estudante(x)]$

B:  $\forall x[Estudante(x) \wedge GostaDe(x, Ana)]$

C:  $\forall x[Estudante(x) \rightarrow GostaDe(x, Ana)]$

D:  $\exists x[Estudante(x) \wedge GostaDe(x, Ana)]$

Resposta: \_\_\_\_

**Resposta:**

C

(b) *A Ana gosta da sua série favorita.*

A:  $GostaDe(Ana, serieFavorita(Ana))$

B:  $GostaDe(Ana, serieFavorita(x))$

C:  $GostaDe(Ana, y) \rightarrow serieFavorita(y, Ana)$

D:  $GostaDe(Ana, y) \wedge serieFavorita(y, Ana)$

Resposta: \_\_\_\_

**Resposta:**

A

3. (1.5) Considere que a seguinte tabela (incompleta) foi usada para calcular o unificador mais geral (UMG) entre duas fórmulas  $P(x, f(y), f(b))$  e  $P(z, f(a), x)$ . Considere ainda que  $P$  é um predicado,  $x, y$  e  $z$  são variáveis,  $a$  e  $b$  são constantes e  $f$  é uma função.

Conjunto de fbfs	Conjunto de desacordo	Substituição
$\{P(x, f(y), f(b)), P(z, f(a), x)\}$		
$\{P(z, f(y), f(b)), P(z, f(a), z)\}$	$\{y, a\}$	
	$\{f(b), z\}$	
$\{P(f(b), f(a), f(b))\}$		

**UMG =**

Preencha a tabela acima e indique explicitamente o UMG (caso exista) entre as fórmulas dadas.

**Resposta:**

Conjunto de fbfs	Conjunto de desacordo	Substituição
$\{P(x, f(y), f(b)), P(z, f(a), x)\}$	$\{x, z\}$	$\{z/x\}$
$\{P(z, f(y), f(b)), P(z, f(a), z)\}$	$\{y, a\}$	$\{a/y\}$
$\{P(z, f(a), f(b)), P(z, f(a), z)\}$	$\{f(b), z\}$	$\{f(b)/z\}$
$\{P(f(b), f(a), f(b))\}$		

**UMG =**  $\{f(b)/x, a/y, f(b)/z\}$

4. (2.0) Demonstre o seguinte argumento, indicando as regras de inferência da Lógica de Primeira Ordem usadas (apenas pode usar as regras de premissa, hipótese, repetição, reiteração, e as regras de introdução e eliminação de cada um dos símbolos lógicos):

$$\{\forall x[(R(x) \wedge S(x))], \forall x[(S(x) \rightarrow Q(x))]\} \vdash \forall x[Q(x)]$$

Para tal, preencha o seguinte esquema de prova:

1	
2	
3	$x_0$
4	
5	
6	
7	
8	
9	

**Resposta:**

1	$\forall x[(R(x) \wedge S(x))]$	Prem
2	$\forall x[(S(x) \rightarrow Q(x))]$	Prem
3	$x_0 \mid \forall x[(S(x) \rightarrow Q(x))]$	Rei, 2
4	$S(x_0) \rightarrow Q(x_0)$	E $\forall$ , 3
5	$\forall x[(R(x) \wedge S(x))]$	Rei, 1
6	$R(x_0) \wedge S(x_0)$	E $\forall$ , 5
7	$S(x_0)$	E $\wedge$ , 6
8	$Q(x_0)$	E $\rightarrow$ , (7, 4)
9	$\forall x[Q(x)]$	I $\forall$ , (3, 8)

5. (2.0) Complete a seguinte tabela, indicando a posição em que as fórmulas apareceriam na sequência de passos que permite transformar a fórmula da linha h (1ª posição na prova) na sua forma clausal (linha b; 8ª e última posição na prova).

Linha	Fórmulas	Posição na conversão
a	$\forall x[\neg A(x) \vee \exists y[B(x, y) \wedge C(y)]] \wedge \exists z[D(z)]$	
b	$\{\{\neg A(x), B(x, f(x))\}, \{\neg A(x), C(f(x))\}, \{D(a)\}\}$	8ª
c	$(\neg A(x) \vee (B(x, f(x)) \wedge C(f(x)))) \wedge D(a)$	
d	$\forall x[\neg A(x) \vee \exists y[B(x, y) \wedge C(y)]] \wedge \exists x[D(x)]$	
e	$(\neg A(x) \vee B(x, f(x))) \wedge (\neg A(x) \vee C(f(x))) \wedge D(a)$	
f	$\forall x[\neg A(x) \vee (B(x, f(x)) \wedge C(f(x)))] \wedge D(a)$	
g	$\{\neg A(x) \vee B(x, f(x)), \neg A(x) \vee C(f(x)), D(a)\}$	
h	$\forall x[A(x) \rightarrow \exists y[B(x, y) \wedge C(y)]] \wedge \exists x[D(x)]$	1ª

**Resposta:**

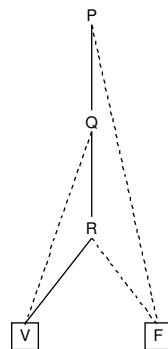
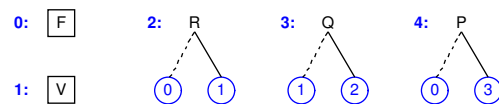
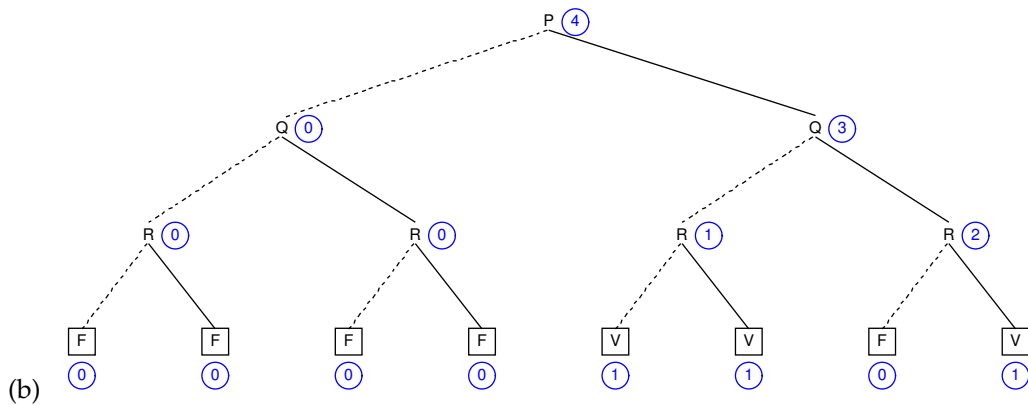
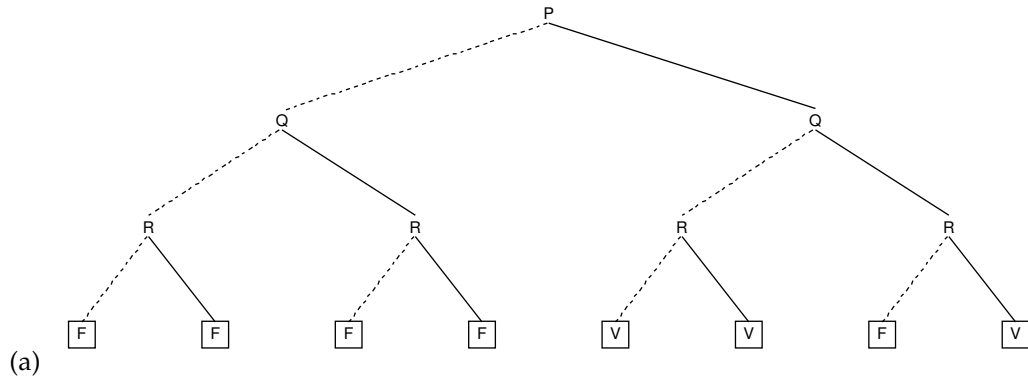
Fórmulas Intermédias	Posição na conversão
$\forall x[\neg A(x) \vee \exists y[B(x, y) \wedge C(y)]] \wedge \exists z[D(z)]$	3ª
$\{\{\neg A(x), B(x, f(x))\}, \{\neg A(x), C(f(x))\}, \{D(a)\}\}$	8ª
$(\neg A(x) \vee (B(x, f(x)) \wedge C(f(x)))) \wedge D(a)$	5ª
$\forall x[\neg A(x) \vee \exists y[B(x, y) \wedge C(y)]] \wedge \exists x[D(x)]$	2ª
$(\neg A(x) \vee B(x, f(x))) \wedge (\neg A(x) \vee C(f(x))) \wedge D(a)$	6ª
$\forall x[\neg A(x) \vee (B(x, f(x)) \wedge C(f(x)))] \wedge D(a)$	4ª
$\{\neg A(x) \vee B(x, f(x)), \neg A(x) \vee C(f(x)), D(a)\}$	7ª
$\forall x[A(x) \rightarrow \exists y[B(x, y) \wedge C(y)]] \wedge \exists x[D(x)]$	1ª

6. (4.0) Considere a *fbf*

$$(P \wedge (Q \rightarrow R))$$

- (a) (1.5) Desenhe a árvore de decisão da *fbf* anterior, usando a ordem alfabética entre os símbolos de proposição.
- (b) (2.5) Obtenha o OBDD reduzido da *fbf* dada, por aplicação dos algoritmos *reduz* e *compacta* à árvore de decisão obtida na alínea a).

**Resposta:**



7. (2.0) Considere o algoritmo DP e considere que já foi feita a seguinte distribuição por baldes:

$b_A: \{\neg A, B, C\}, \{\neg A, D\}$

$b_B: \{B\}$

$b_C: \{C, D\}$

$b_D:$

- (a) (0.5) Sem fazer cálculos, indique a fórmula (forma clausal) que originou este preenchimento de baldes.

**Resposta:**

$\{\{\neg A, B, C\}, \{\neg A, D\}, \{B\}, \{C, D\}\}$

- (b) (0.5) Faça o processamento de baldes, indicando as fórmulas obtidas. No caso de não existirem, diga-o explicitamente.

**Resposta:**

Nada a fazer. Fica exactamente como está.

- (c) (0.5) Das seguintes frases indique a única que **NÃO** é verdade:

A: é possível obter uma testemunha em que D é verdadeiro e B é falso

B: é possível obter uma testemunha em que D é verdadeiro e B é verdadeiro

C: é possível obter uma testemunha em que D é verdadeiro e C é falso

D: é possível obter uma testemunha em que D é verdadeiro e C é verdadeiro

Resposta: \_\_\_\_

**Resposta:**

A

- (d) (0.5) Indique qual das seguintes interpretações representa uma testemunha:

A:  $I(D) = F, I(C) = F, I(B) = F, I(A) = F$

B:  $I(D) = F, I(C) = F, I(B) = V, I(A) = V$

C:  $I(D) = V, I(C) = V, I(B) = F, I(A) = F$

D:  $I(D) = V, I(C) = V, I(B) = V, I(A) = F$

Resposta: \_\_\_\_

**Resposta:**

D

8. (1.0) Considere a conceptualização  $(D, F, R)$  em que:

$$D = \{\diamond, \square\}$$

$$F = \{\}$$

$$R = \{\dots\}.$$

Considere a interpretação  $I: \{a, b, P, S\} \mapsto D \cup F \cup R$ , tal que:

$$I(a) = \diamond$$

$$I(b) = \square$$

Preencha a tabela abaixo, de forma a que a interpretação  $I$  seja um modelo do conjunto de *fbfs*

$$\Delta = \{\neg P(a), P(a) \vee \neg P(b), \forall x[P(x) \vee S(x)]\}.$$

$I(P)$	
$I(S)$	

**Resposta:**

$I(P)$	$\{\}$
$I(S)$	$\{(\diamond), (\square)\}$

9. (1.5) Assuma definido o predicado `primo(N)` que é verdade se `N` for um número primo. Considere ainda o predicado `soPrimos(L1, L2)` que é verdade se `L1` for uma lista com inteiros e `L2` a lista obtida a partir de `L1` por eliminação dos inteiros não primos. No caso de não existir nenhum primo em `L1`, `L2` será a lista vazia. Por exemplo:

```
?- L1 = [1, 2, 8, 5, 4, 12], soPrimos(L1, L2).
L2 = [1, 2, 5].
```

Implemente uma versão iterativa do predicado `soPrimos(L1, L2)`. Sugestão: use o predicado `primo/1`.

**Resposta:**

```
soPrimos(L1, L2) :- soPrimos(L1, L2, []).

soPrimos([], L2, L2).

soPrimos([N | Cauda], L2, Aux) :-
    primo(N), !,
    append(Aux, [N], Aux1),
    soPrimos(Cauda, L2, Aux1).

soPrimos([_ | Cauda], L2, Aux) :-
    soPrimos(Cauda, L2, Aux).
```

10. (0.5 + 0.5 + 0.5) Considere o seguinte programa em Prolog:

```
C1: le(X, Y) :- estudante(X), livro(Y).
C2: estudante(maria).
C3: livro(lotr).
C4: livro(diarioDeUmBanana).
```

Supondo que vão sempre ser pedidas mais respostas enquanto tal for possível, qual a resposta do Prolog ao objectivo:

```
?- le(X, Y).
```

- (a) ... considerando o programa anterior.

Resposta:

**Resposta:**

```
X = maria, Y = lotr ; X = maria, Y = diarioDeUmBanana.
```

- (b) ... considerando que `C1` é agora:

```
le(X, Y) :- estudante(X), livro(Y), !.
```

Resposta:

**Resposta:**

```
X = maria, Y = lotr
```

- (c) ... considerando que `C1` é agora:

```
le(X, Y) :- estudante(X), \+(livro(Y)).
```

Resposta:

**Resposta:**

```
false.
```



11. (a) (0.75) Implemente o predicado `divideTres/4`, tal que `divideTres(L, L1, L2, L3)` significa que as listas `L1`, `L2` e `L3`, concatenadas por esta ordem, resultam na lista `L`. Por exemplo, supondo que se pedem todas as respostas:

```
?- divideTres([a, b, c, d], L1, L2, L3).
L1 = L2, L2 = [],
L3 = [a, b, c, d] ;
L1 = [],
L2 = [a],
L3 = [b, c, d] ;
L1 = [],
L2 = [a, b],
L3 = [c, d] ;
L1 = [],
L2 = [a, b, c],
L3 = [d] ;
L1 = L3, L3 = [],
L2 = [a, b, c, d] ;
...
L1 = [a, b],
L2 = [c],
L3 = [d] ;
L1 = [a, b],
L2 = [c, d],
L3 = [] ;
L1 = [a, b, c],
L2 = [],
L3 = [d] ;
L1 = [a, b, c],
L2 = [d],
L3 = [] ;
L1 = [a, b, c, d],
L2 = L3, L3 = [] ;
false.
```

Sugestão: utilize o predicado `append`.

**Resposta:**

```
divideTres(L, L1, L2, L3) :-
    append([L1, L2, L3], L).
```

- (b) (0.75) Implemente o predicado `sublista_central/2`, tal que `sublista_central(Lst, SL)` significa que `SL` é uma sublista central da lista `Lst`. Uma sublista `SL`, de uma lista `Lst`, é central, se existir o mesmo número de elementos antes e depois da sublista `SL`, na lista `Lst`. Por exemplo, supondo que se pedem todas as respostas:

```
?- sublista_central([a,b,c,d], SL).
SL = [a, b, c, d];
SL = [b, c];
SL = [];
false.
```

```
?- sublista_central([a,b,c,d,e], SL).
SL = [a, b, c, d, e];
SL = [b, c, d];
SL = [c];
false.
```

Sugestão: utilize o predicado definido anteriormente.

**Resposta:**

```
sublista_central(Lst, SLCentral) :-
    divideTres(Lst, Inicio, SLCentral, Fim),
    length(Inicio, Comp),
    length(Fim, Comp).
```

- (c) (1.0) Usando o predicado definido na alínea anterior, implemente o predicado `sublistas_centrais/2`, tal que `sublistas_centrais(Lst, SLs)` significa que `SLs` é a lista de todas as sublistas centrais da lista `Lst`. Exemplo:

```
?- sublistas_centrais([a,b,c,d], SLs).
SLs = [[a, b, c, d], [b, c], []].
```

```
?- sublistas_centrais([a,b,c,d,e], SLs).
SLs = [[a, b, c, d, e], [b, c, d], [c]].
```

Pode assumir que a lista dada não contém variáveis. Pode, ou não, usar meta-predicados. Fica ao seu critério.

**Resposta:**

```
sublistas_centrais(Lst, SLs) :-
    findall(SL, sublista_central(Lst, SL), SLs).
```

RASCUNHO

RASCUNHO