Lógica para Programação

Solução do Exame de 1ª Época

9 de Junho de 2016

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) Numa lógica correcta nenhum argumento não válido é demonstrável.
   Resposta: ___
   Resposta:
   V

   (b) Uma interpretação não é um modelo de um conjunto de *fbfs*, se e só se não satisfizer nenhuma *fbf* do conjunto.
   Resposta: ___
   Resposta:
   F

2. Considere os seguintes predicados:

$$\begin{aligned} Inteiro(x) &= x \text{ é um número inteiro} \\ Natural(x) &= x \text{ é um número natural} \\ Par(x) &= x \text{ é um número par} \\ \acute{I}mpar(x) &= x \text{ é um número ímpar} \\ Dobro(x, y) &= y \text{ é o dobro de } x \end{aligned}$$

   Represente em Lógica de Primeira Ordem as seguintes proposições:

   (a) (0.5) Todos os naturais são inteiros, mas nem todos os inteiros são naturais.

   (b) (0.5) Para qualquer inteiro ímpar, existe um inteiro par que é o dobro desse inteiro ímpar.

   Resposta:

   (a) $\forall x[Natural(x) \rightarrow Inteiro(x)] \wedge \exists x[Inteiro(x) \wedge \neg Natural(x)]$ ou
   $\forall x[Natural(x) \rightarrow Inteiro(x)] \wedge \neg\forall x[Inteiro(x) \rightarrow Natural(x)]$

   (b) $\forall x[(Inteiro(x) \wedge \acute{I}mpar(x)) \rightarrow \exists y[Inteiro(y) \wedge Par(y) \wedge Dobro(x, y)]]$

3. (1.5) Considere o seguinte conjunto de *fbfs* (em que x e y são variáveis e a é uma constante)

$$\{P(x, y), P(y, a)\}$$

Preencha as linhas necessárias da seguinte tabela, de forma a seguir o algoritmo de unificação para determinar se as *fbfs* são unificáveis. Em caso afirmativo, indique o unificador mais geral; caso contrário, indique que as *fbfs* não são unificáveis.

Conjunto de fbfs	**Conjunto de desacordo**	**Substituição**

Unificador mais geral (se existir):

Resposta:

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

Unificador mais geral (se existir): $\{x/y\} \circ \{a/x\} = \{a/y, a/x\}$

4. (2.0) Demonstre que

$$\{\forall x[P(x) \to R(x)] \lor \forall x[Q(x) \to R(x)]\} \vdash \forall x[(P(x) \land Q(x)) \to R(x)]$$

usando o sistema dedutivo da Lógica de Primeira Ordem (apenas pode usar as regras de premissa, hipótese, repetição, reiteração, e as regras de introdução e eliminação de cada um dos símbolos lógicos).

Resposta:

1		$\forall x[P(x) \to R(x)] \vee \forall x[Q(x) \to R(x)]$	Prem
2	x_0	$P(x_0) \wedge Q(x_0)$	Hip
3		$P(x_0)$	E$\wedge$, 2
4		$Q(x_0)$	E$\wedge$, 2
5		$\forall x[P(x) \to R(x)] \vee \forall x[Q(x) \to R(x)]$	Rei, 1
6		$\forall x[P(x) \to R(x)]$	Hip
7		$P(x_0)$	Rei, 3
8		$P(x_0) \to R(x_0)$	E$\forall$, 6
9		$R(x_0)$	E$\to$, 7, 8
10		$\forall x[Q(x) \to R(x)]$	Hip
11		$Q(x_0)$	Rei, 4
12		$Q(x_0) \to R(x_0)$	E$\forall$, 10
13		$R(x_0)$	E$\to$, 11, 12
14		$R(x_0)$	E$\vee$, (5, (6, 9), (10, 13))
15		$(P(x_0) \wedge Q(x_0)) \to R(x_0)$	I$\to$, (2, 14)
16		$\forall x[(P(x) \wedge Q(x)) \to R(x)]$	I$\forall$, (2, 15)

5. (2.0) Demonstre o seguinte teorema

$$\exists x[P(x) \wedge Q(x)] \to (\exists x[P(x)] \wedge \exists x[Q(x)])$$

usando resolução.

Resposta:

Como se trata de um teorema, vamos passar a sua negação à forma clausal, e fazer uma prova por refutação:

- *Passagem à forma clausal:*
  $\neg(\exists x[P(x) \wedge Q(x)] \to (\exists x[P(x)] \wedge \exists x[Q(x)]))$
  $\neg(\neg\exists x[P(x) \wedge Q(x)] \vee (\exists x[P(x)] \wedge \exists x[Q(x)]))$
  $\exists x[P(x) \wedge Q(x)] \wedge \neg(\exists x[P(x)] \wedge \exists x[Q(x)])$
  $\exists x[P(x) \wedge Q(x)] \wedge (\neg\exists x[P(x)] \vee \neg\exists x[Q(x)])$
  $\exists x[P(x) \wedge Q(x)] \wedge (\forall x[\neg P(x)] \vee \forall x[\neg Q(x)])$
  $\exists x[P(x) \wedge Q(x)] \wedge (\forall y[\neg P(y)] \vee \forall z[\neg Q(z)])$
  $P(a) \wedge Q(a) \wedge (\forall y[\neg P(y)] \vee \forall z[\neg Q(z)])$
  (em que a é uma constante de Skolem)
  $P(a) \wedge Q(a) \wedge (\neg P(y) \vee \neg Q(z))$
  $\{\{(P(a)\}, \{Q(a)\}, \{\neg P(y), \neg Q(z)\}\}$

- *Prova:*

1	$\{P(a)\}$	Prem
2	$\{Q(a)\}$	Prem
3	$\{\neg P(y), \neg Q(z))\}$	Prem
4	$\{\neg Q(z)\}$	Res, (1,3), $\{a/y\}$
5	$\{\}$	Res, (2,4), $\{a/z\}$

6. (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*

   $$\Delta = \{P(a), \neg P(b), S(a), S(b)\}.$$

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

   Resposta:

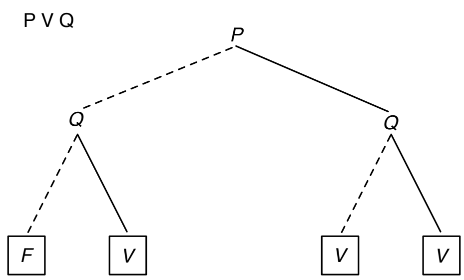
$I(P)$	$\{(\diamond)\}$
$I(S)$	$\{(\diamond), (\square)\}$

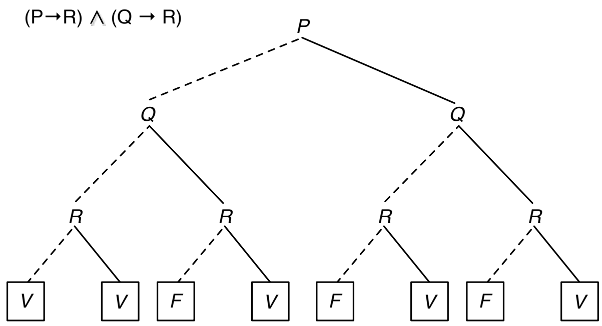
7. Usando OBDDs, prove que

   $$\{P \vee Q, (P \to R) \wedge (Q \to R)\} \models R$$

   seguindo os seguintes passos:

   (a) (0.5) Desenhe as árvores de decisão de cada uma das premissas, usando a ordem $P \prec Q \prec R$.
   Resposta:





(b) (1.0) Aplique o algoritmo *reduz* a cada uma das árvores obtidas na alínea anterior.

Resposta:

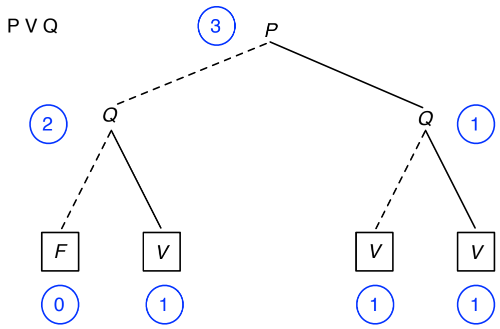
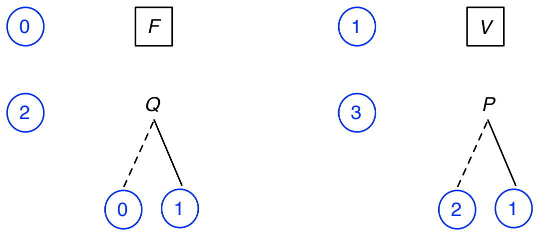


Tabela associativa:



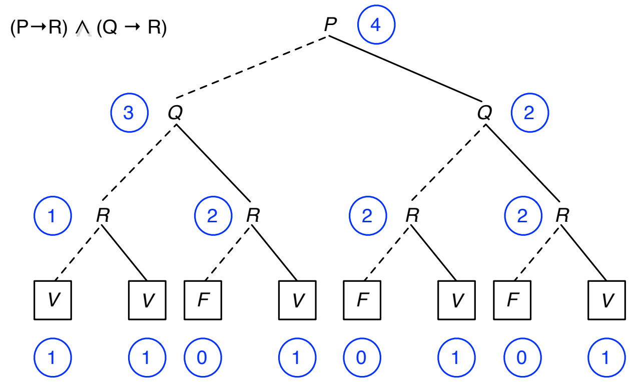
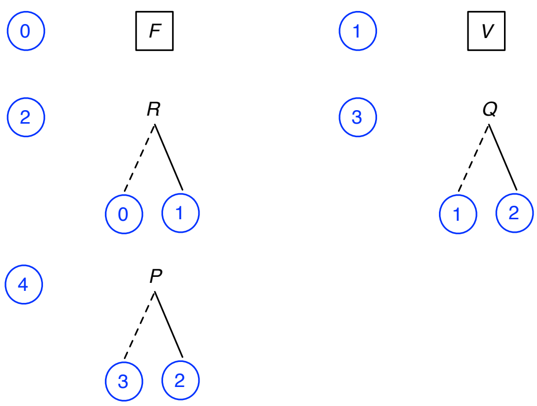
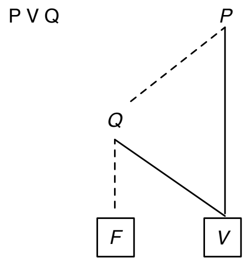


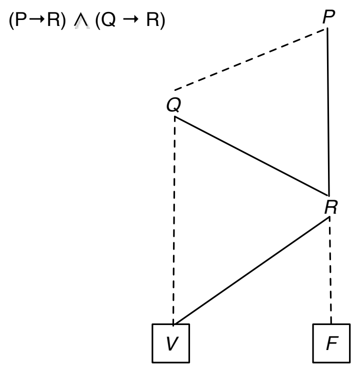
Tabela associativa:



(c) (1.0) Aplique o algoritmo *compacta* aos resultados obtidos na alínea anterior.

Resposta:





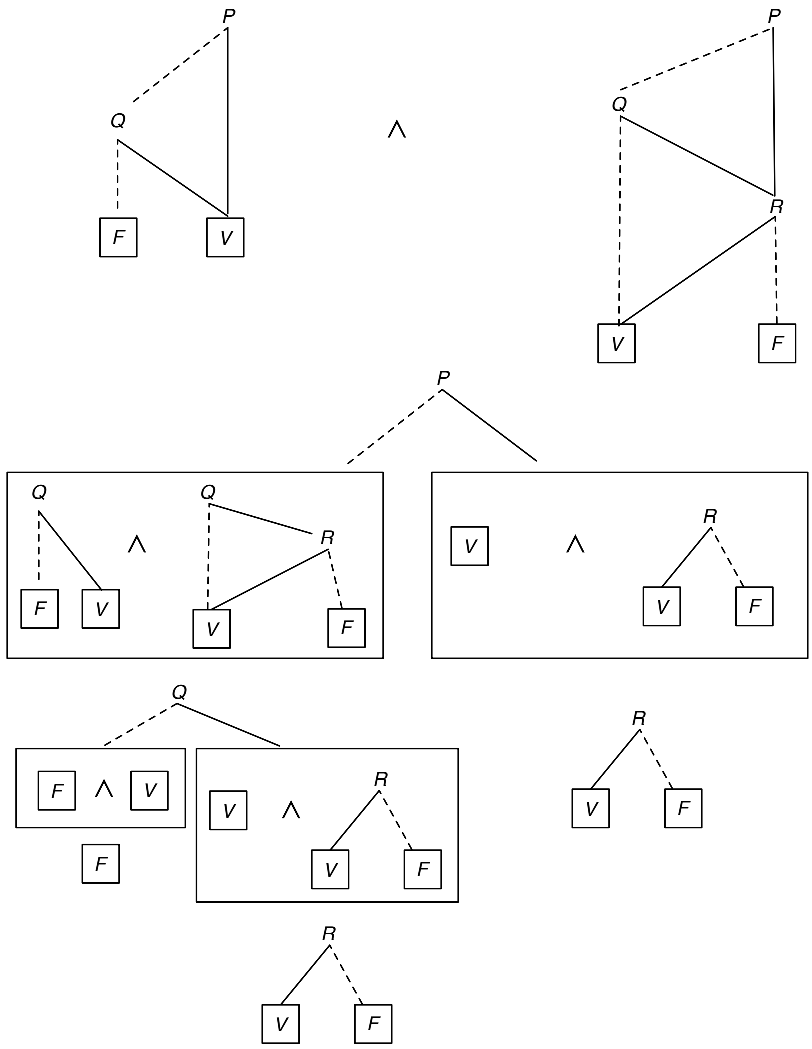
(d) (1.0) Aplique o algoritmo $aplica$ aos OBDDs reduzidos das premissas, e ao *símbolo lógico adequado para o fim em vista*.

Resposta:

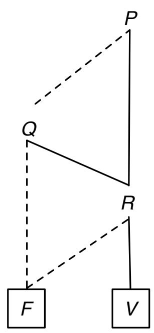
Uma vez que estamos interessados em determinar os modelos das premissas, temos de determinar os modelos da *fbf*

$$(P \vee Q) \wedge ((P \to R) \wedge (Q \to R)).$$

Assim, o símbolo lógico adequado é a conjunção.



Após redução:



(e) (0.5) Usando o resultado da alínea anterior, indique os modelos das premissas. Pode concluir que

$$\{P \vee Q, (P \to R) \wedge (Q \to R)\} \models R?$$

Justifique a sua resposta.

Resposta:

$$\begin{array}{lll} M_1(P) = V & M_1(Q) = V & M_1(R) = V \\ M_2(P) = V & M_2(Q) = F & M_2(R) = V \\ M_3(P) = F & M_3(Q) = V & M_3(R) = V \end{array}$$

Sim, porque a conclusão R é verdadeira em todos os modelos das premissas.

8. (2.0) Considere o seguinte conjunto de *fbfs* :

$$\{(P \to R) \vee (Q \to R), P, Q, R\}$$

Estabelecendo a ordem correspondente à ordem alfabética entre os símbolos de proposição, use o algoritmo DP recorrendo a baldes para determinar se o conjunto é satisfazível. Em caso afirmativo apresente uma testemunha.

Resposta:

(a) *Passagem à forma clausal:*
$\{\{\neg P, R, \neg Q\}, \{P\}, \{Q\}, \{R\}\}$

(b) *Criação e preenchimento dos baldes:*
$b_P: \ \{\neg P, R, \neg Q\}, \{P\}$
$b_Q: \ \{Q\}$
$b_R: \ \{R\}$

(c) *Processamento dos baldes:*
$b_P: \ \{\neg P, R, \neg Q\}, \{P\}$
$b_Q: \ \{Q\} \qquad \{R, \neg Q\}$
$b_R: \ \{R\} \qquad \{R\}$
Como não foi gerada a cláusula vazia, o conjunto é satisfazível.

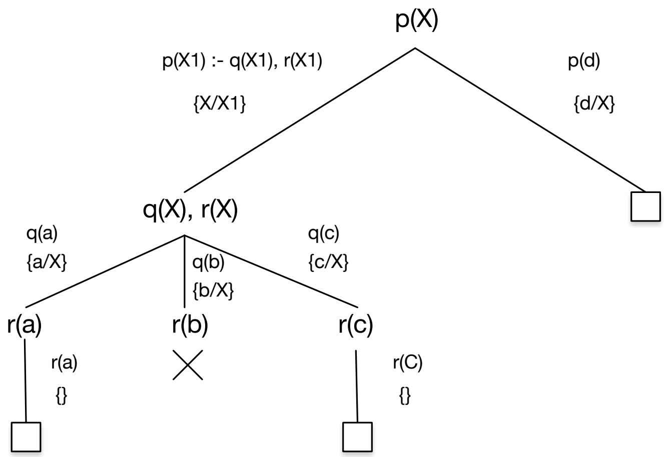
(d) *Inspeção dos baldes:*
$I(R) = V$ (obrigatório)
$I(Q) = V$ (obrigatório)
$I(P) = V$ (obrigatório)

9. Considere o seguinte programa em PROLOG :

```
p(X) :- q(X),r(X).
p(d).
q(a).
q(b).
q(c).
r(a).
r(c).
```

(a) (1.0) Suponha que é dado o objectivo `?- p(X)` ao PROLOG , e que são pedidas todas as respostas. Desenhe a árvore SLD gerada. Em cada ramo indique a cláusula usada, e respectiva substituição.

Resposta:



(b) (0.5) Altere o programa anterior de modo a obter apenas a primeira e a última respostas.

Resposta:

A afirmação `q(a).` deve ser substituída por `q(a) :- !.`

10. (a) (1.0) Defina o predicado `par/1`, tal que `par(N)`, em que `N` é um inteiro, significa que `N` é par. Usando este predicado defina o predicado `filtra_pares/2`, tal que `filtra_pares(L1, L2)`, em que `L1` e `L2` são listas de inteiros, significa que `L2` é constituída pelos elementos pares de `L1`. Por exemplo:

```
?- filtra_pares([2,3,4,5,6],L).
L = [2, 4, 6].

?- filtra_pares([2,3,4,5,6],[4,6]).
false.
```

Resposta:

```
par(X) :- X mod 2 =:= 0.

filtra_pares([], []) :- !.

filtra_pares([P1 | R1], L2) :-
    par(P1),
    !,
    filtra_pares(R1, R),
    L2 = [P1 | R].

filtra_pares([_ | R1], R) :-
    filtra_pares(R1, R).
```

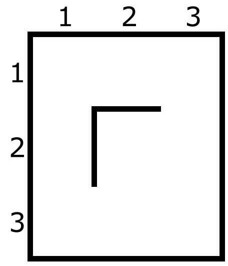
(b) (0.5) Considere definido o predicado `retira(L1, L2, L3)`, em que `L3` é a lista resultante de retirar de `L1` os elementos de `L2` . Explique sucintamente o que faz o predicado `xpto/2`, em que `L1` e `L2` são listas de inteiros, definido como se segue:

```
xpto(L1, L2)  :- filtra_pares(L1, L3), retira(L1, L3, L2).
```

Resposta:

Retira os elementos pares de uma lista.

11. (a) (1.5) Considere que está a implementar uma variante do projecto, em que, para além dos movimentos na horizontal e na vertical, também é possível fazer movimentos na diagonal. Um movimento na diagonal é representado por uma lista de dois elementos em que um deles é `c` ou `b`, e o outro é `e` ou `d`, por exemplo `[c, e]`. Um movimento na diagonal, `[D1, D2]` é possível, se for possível efectuar o movimento na direcção `D1`, seguido do movimento na direcção `D2`, independentemente das posições visitadas anteriormente.



Por exemplo, dado o labirinto representado acima, a partir da posição `(2, 2)` são possíveis os seguintes movimentos na diagonal:

- `[d, c]` para a posição `(1, 3)`
- `[b, d]` e `[d, b]` para a posição `(3, 3)`
- `[b, e]` para a posição `(3, 1)`

Defina o predicado `diag_possivel/4`, tal que `diag_possivel(Lab, Diag, Pos_actual, Pos_nova)` significa que, dado o labirinto `Lab`, é possível o movimento na diagonal `Diag`, a partir da posição `Pos_actual`, conduzindo à posição `Pos_nova`. Por exemplo, sendo `Lab` o labirinto da figura, temos:

```
?- ..., diag_possivel(Lab, [c, d], (2, 2), Pos_nova).
false.

?- ..., diag_possivel(Lab, [d, c], (2, 2), Pos_nova).
Lab = ...,
Pos_nova = (1, 3)
```

Sugestão: utilize o predicado, que definiu no projecto,
`movs_possiveis(Lab, Pos_actual, Movs_anteriores, Poss)`.

Resposta:

```
diag_possivel(Lab, [D1, D2], Pos_actual, (L, C)) :-
     movs_possiveis(Lab, Pos_actual, [], Poss1),
     member((D1, L1, C1), Poss1),
     movs_possiveis(Lab,  (L1, C1), [], Poss2),
     member((D2, L, C), Poss2).
```

(b) (1.0) Usando o predicado definido na alínea anterior, defina o predicado `diag_possivel_reverte/4`, tal que
`diag_possivel_reverte(Lab, [D1, D2], Pos_actual, Pos _nova)`

significa que, dado o labirinto `Lab`, é possível o movimento na diagonal `[D1, D2]` ou `[D2, D1]`, a partir da posição `Pos_actual`, conduzindo à posição `Pos_nova`. Por exemplo, sendo `Lab` o labirinto da figura, temos:

```
?- ..., diag_possivel_reverte(Lab, [c, d], (2, 2), Pos_nova).
Lab = ...,
Pos_nova = (1, 3)

?- ..., diag_possivel_reverte(Lab, [c, e], (2, 2), Pos_nova).
false.
```

Resposta:

```
diag_possivel_reverte(Lab, [D1, D2], Pos_actual, Pos_nova) :-
     diag_possivel(Lab, [D1, D2], Pos_actual, Pos_nova), !
                          ;
     diag_possivel(Lab, [D2, D1], Pos_actual, Pos_nova).
```