Lógica para Programação

Segundo Exame

2 de Julho de 2015

15:00–17:00

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

   (a) Numa lógica não completa, nenhum argumento válido é demonstrável.
   Resposta: ______
   Resposta:
   Falsa

   (b) Se (Δ, α) é um contra-argumento para o argumento (Δ', α'), então os dois argumentos têm a mesma forma.
   Resposta: ______
   Resposta:
   Verdadeira

2. **(1.5)** Complete cada uma das seguintes afirmações. Cada resposta certa vale 0.5 valores.

   (a) Numa lógica correta, todos os argumentos demonstráveis são ______________.
   Resposta:
   Válidos

   (b) Uma lógica cujo sistema dedutivo permita derivar todas as fórmulas bem formadas é ______________.
   Resposta:
   Completa

   (c) Supondo que o argumento (Δ, α) é válido numa determinada lógica que é ______________, então $\Delta \vdash \alpha$.
   Resposta:
   Completa

3. **(2.0)** Escolha a *única* resposta *correcta* para as seguintes questões. Cada resposta certa vale 1 valor e *cada resposta errada desconta 0.4 valores*.

   (a) A base de Herbrand para o conjunto de cláusuas $\{\{P(a)\}, \{Q(f(b))\}\}$ é:

   A. $\{a, b, f(a), f(b), f(f(a)), f(f(b)), \ldots\}$
   B. $\{P(a), Q(a), P(f(a)), Q(f(b)), P(f(f(a))), Q(f(f(b))), \ldots\}$

C. $\{a, b\}$
D. $\{P(a), Q(b)\}$

Resposta: ______

Resposta:

B.

(b) Para as cláusulas de Horn

$$Ant(x,y) \leftarrow AD(x,y) \tag{1}$$

$$Ant(x,z) \leftarrow Ant(x,y),\ AD(y,z) \tag{2}$$

$$AD(Marge, Bart) \leftarrow \tag{3}$$

$$AD(Sr.B, Marge) \leftarrow \tag{4}$$

$$\leftarrow Ant(Sr.B, Bart) \tag{5}$$

A. (1) e (2) são regras.
B. (5) é uma afirmação.
C. (3) e (4) são objectivos.
D. Nenhuma das afirmações anteriores está correcta.

Resposta: ______

Resposta:

A.

4. **(1.5)** Demonstre o seguinte teorema usando o sistema de dedução natural da lógica proposicional (apenas pode utilizar as regras Hip, Prem, Rep, Reit e introdução e eliminação de cada uma das conectivas):

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

Resposta:

1	$\mid (P \vee Q) \wedge (\neg P \vee R)$	Hip
2	$\mid P \vee Q$	E$\wedge$, 1
3	$\mid \neg P \vee R$	E$\wedge$, 1
4	$\mid \; \mid P$	Hip
5	$\mid \; \mid \neg P \vee R$	Rei, 3
6	$\mid \; \mid \; \mid \neg P$	Hip
7	$\mid \; \mid \; \mid \; \mid \neg R$	Hip
8	$\mid \; \mid \; \mid \; \mid P$	Rei, 4
9	$\mid \; \mid \; \mid \; \mid \neg P$	Rei, 6
10	$\mid \; \mid \; \mid \neg\neg R$	I$\neg$, (7, (8, 9))
11	$\mid \; \mid \; \mid R$	E$\neg$, 10
12	$\mid \; \mid \; \mid Q \vee R$	I$\vee$, 11
13	$\mid \; \mid \; \mid R$	Hip
14	$\mid \; \mid \; \mid Q \vee R$	I$\vee$, 13
15	$\mid \; \mid Q \vee R$	E$\vee$, (5, (6, 12), (13, 14))
16	$\mid \; \mid Q$	Hip
17	$\mid \; \mid Q \vee R$	I$\vee$, 16
18	$\mid Q \vee R$	E$\vee$, (2, (4, 15), (16, 17))
19	$((P \vee Q) \wedge (\neg P \vee R)) \rightarrow (Q \vee R)$	I$\rightarrow$, (1, 17)

5. **(1.5)** Demonstre o seguinte teorema usando o sistema de dedução natural da lógica de primeira ordem (apenas pode utilizar as regras Hip, Prem, Rep, Reit e introdução e eliminação de cada uma das conectivas e quantificadores):

$$(\forall x\ [P(x) \to R(x)] \wedge \exists y\ [P(y)]) \to \exists z[R(z)]$$

Resposta:

1		$\forall x\ [P(x) \to R(x)] \wedge \exists y\ [P(y)]$	Hip
2		$\forall x\ [P(x) \to R(x)]$	E$\wedge$, 1
3		$\exists y\ [P(y)]$	E$\wedge$, 1
4	x_0	$P(x_0)$	Hip
5		$\forall x\ [P(x) \to R(x)]$	Rei, 2
6		$P(x_0)$	Rep, 4
7		$P(x_0) \to R(x_0)$	E$\forall$, 5
8		$R(x_0)$	E$\to$, (6, 7)
9		$\exists z[R(z)]$	I$\exists$, 8
10		$\exists z[R(z)]$	E$\exists$, (3, (4, 9))
11		$(\forall x\ [P(x) \to R(x)] \wedge \exists y\ [P(y)]) \to \exists z[R(z)]$	I$\to$, (1, 11)

6. Considere que α, β, γ e δ são proposições. Sabendo que o argumento

α
β
$\therefore\ \neg\gamma$

é válido, diga, justificando, o que se pode concluir sobre a validade dos seguintes argumentos (válido, não válido, ou não se pode concluir nada):

(a) **(0.5)** α
β
$\therefore\ \gamma \to \delta$

Resposta:

Da validade do argumento original podemos concluir que é impossível ter α e β verdadeiras e γ verdadeira. Podemos assim concluir que sendo α e β verdadeiras o antecedente da implicação $\gamma \to \delta$ é sempre falso, pelo que a implicação é sempre verdadeira. Logo o argumento é válido.

(b) **(0.5)** α
β
γ
$\therefore\ \delta$

Resposta:

Da validade do argumento original podemos concluir que é impossível ter α e β verdadeiras e γ verdadeira. Podemos assim concluir que é impossível ter α, β e γ verdadeiras e δ falsa, logo o argumento é válido.

7. **(1.5)** Considere os seguintes passos de resolução (em Lógica Proposicional) que podem ou não estar correctos.

1	$\{\neg P, S, R\}$	Prem
2	$\{P, Q\}$	Prem
3	$\{\neg Q, \neg R\}$	Prem
4	$\{\neg S\}$	Prem
5	$\{\neg P, R\}$	Res, (1,4)
6	$\{\neg P, \neg Q\}$	Res, (3,5)
7	$\{\}$	Res, (2,6)

Responda às seguintes questões usando apenas a palavra "SIM", a palavra "NÃO", ou um dígito referente ao número da linha em questão. Cada resposta certa vale 0.5 valores e *cada resposta errada desconta 0.2 valores*.

(a) Existe ou não uma linha contendo um passo de resolução incorreto? Em caso afirmativo, indique o número da linha.
Resposta: ______
Resposta:
SIM, 7

(b) Foi ou não seguida uma estratégia de resolução linear?
Resposta: ______
Resposta:
SIM

(c) Foi ou não seguida uma estratégia de resolução unitária?
Resposta: ______
Resposta:
NÃO

8. **(1.5)** Considere o seguinte conjunto de *fbfs*:

$$\{P(x, f(b), y), P(a, f(x), c)\}$$

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

Conjunto de fbfs	**Conj. de Desacordo**	**Substituição**

Unificador mais geral (se existir):

Resposta:

Conjunto de fbfs	Conj. de Desacordo	Substituição
$\{P(x, f(b), y), P(a, f(x), c)\}$	$\{x, a\}$	$\{a/x\}$
$\{P(a, f(b), y), P(a, f(a), c)\}$	$\{b, a\}$	$-$

Unificador mais geral (se existir): As *fbfs* não são unificáveis.

9. Considere a *fbf* $\{\{\neg P, Q, R\}, \{\neg Q, R, S\}, \{\neg R, S\}, \{\neg P, \neg Q, R\}, \{\neg S\}, \{Q, \neg R\}\}$ e o algoritmo DP implementado recorrendo a baldes.

   (a) **(1.0)** Complete a tabela que se segue:

Baldes	Cláusulas originais	Resolventes
b_P		
b_Q		
b_R		
b_S		

   (b) **(0.5)** Conclui que a *fbf* é satisfazível ou não satisfazível? Justifique. No caso de a *fbf* ser satisfazível indique uma testemunha.

Resposta:

(a)

Baldes	Cláusulas originais	Resolventes
b_P	$\{\neg P, Q, R\}$ $\{\neg P, \neg Q, R\}$	
b_Q	$\{\neg Q, R, S\}$ $\{Q, \neg R\}$	
b_R	$\{\neg R, S\}$	
b_S	$\{\neg S\}$	

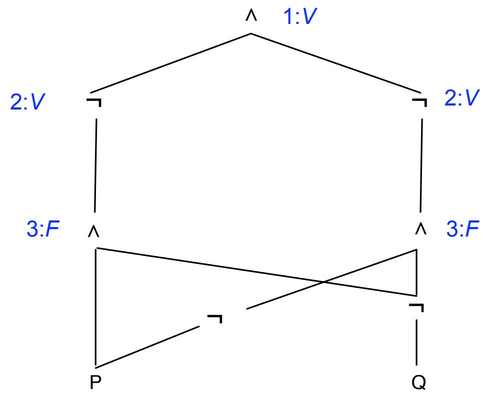
(b) A *fbf* é satisfazível porque não foi gerada a cláusula vazia durante o processamento dos baldes. A única testemunha é: $I(P) = I(Q) = I(R) = I(S) = F$.

10. **(1.5)** Use o algoritmo de propagação de marcas para determinar se a *fbf* $(P \to Q) \wedge (\neg P \to Q)$ é satisfazível. Em caso afirmativo apresente uma testemunha.

Resposta:

(a) Eliminação do símbolo $\to$: $\neg(P \wedge \neg Q) \wedge (\neg P \wedge \neg Q)$.

(b) Obtenção do DAG da *fbf* $\neg(P \wedge \neg Q) \wedge (\neg P \wedge \neg Q)$ e propagação de marcas:



(c) Como os nós P e Q ficaram por marcar, é aplicado o algoritmo de teste de nós. Apresentam-se as várias alternativas (uma delas seria suficiente):

- O teste do nó P com a marca V, marca o nó Q com V. Obtém-se uma marcação completa e consistente. Assim, uma testemunha é: $I(P) = V$, $I(Q) = V$.

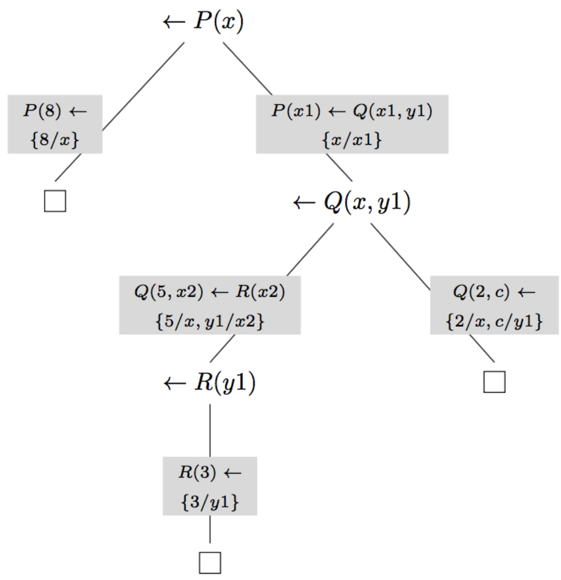
- O teste do nó P com a marca F, marca o nó Q com V. Obtém-se uma marcação completa e consistente. Assim, uma testemunha é: $I(P) = F$, $I(Q) = V$.

- O teste do nó Q com a marca V, deixa o nó P por marcar. Assim, o nó Q seria agora testado com a marca F. Este teste leva a uma contradição. Logo, o nó Q é marcado com a marca permanente V. A propagação desta marca não consegue marcar P. Logo, é novamente aplicado o algoritmo de teste de nós. O teste de P com qualquer marca leva a uma marcação completa e consistente. Assim, obteríamos uma das duas testemunhas: $I(P) = V$, $I(Q) = V$ ou $I(P) = F$, $I(Q) = V$.

11. **(1.5)** Usando uma árvore de resolução SLD e uma função de selecção que escolhe o primeiro literal do objectivo para unificar, indique explicitamente todas as soluções para o objectivo $\leftarrow P(x)$. (Em cada ramo da árvore indique a cláusula e substituição respectivas.)

$P(8)$.
$P(x) \leftarrow Q(x, y)$.
$Q(5, x) \leftarrow R(x)$.
$Q(2, c)$.
$R(3)$.

Resposta:



As soluções são: $\{8/x\}$, $\{5/x\}$ e $\{2/x\}$.

12. **(1.0)** Considere o predicado `penultimo/2` tal que `penultimo(L,X)` afirma que `X` é o penúltimo elemento da lista `L`.

    Por exemplo:

```
?- penultimo([a,b,c,d],X).
X = c
```

    Complete o código que se segue:

```
penultimo(            , X).
penultimo(            , X)  :- penultimo([Y|Ys],          ).
```

    Resposta:

```
penultimo([X,_], X).
penultimo([_,Y|Ys], X) :- penultimo([Y|Ys], X).
```

13. **(1.0)** Considere o predicado `retira_N/3` tal que `retira_N(L1,N,L2)` afirma que `L1` é uma lista, `N` é um inteiro e `L2` corresponde à lista `L1` em que o `N`-ésimo elemento foi retirado. (O primeiro elemento da lista corresponde a `N=1`.)

    Por exemplo:

```
?- retira_N ([a,b,c,d], 3, L).
L = [a,b,d]
```

    Complete o código que se segue:

```
retira_N(L1, N, L2) :- retira_N(L1, N, L2, N).
```

```
retira_N([], _,       , _).
retira_N([_|Xs], N, Ys, 1) :-
retira_N(        , N,       , K) :- K > 1,
                                    K1 is      ,
                                    retira_N(Xs, N, Ys, K1).
```

Resposta:

```
rretira_N(L1, N, L2) :- retira_N(L1, N, L2, N).

retira_N([], _, [], _).
retira_N([_|Xs], N, Ys, 1) :- Xs = Ys.
retira_N([X|Xs], N, [X|Ys], K) :- K > 1, K1 is K - 1, retira_N(Xs, N, Ys, K1).
```

14. **(1.0)** Considere o seguinte programa em PROLOG:

```
serie('Game of Thrones').
serie(galactica).
serie('csi NY').
policial('csi NY').
fantasia('Game of Thrones').
ficcaoCientifica(galactica).
gosta1('Alberto',S):-serie(S),\+policial(S).
gosta2('Alberto',S):-\+fantasia(S),serie(S).
gosta3('Alberto',S):-\+documentario(S),serie(S).
```

Indique todos os valores devolvidos para os objectivos `gosta1(X,Y)`,`gosta2(X,Y)` e `gosta3(X,Y)`.

Resposta:

```
?-gosta1(X,Y).
X='Alberto',
Y='Game of Thrones';
X='Alberto',
Y=galactica.

?-gosta2(X,Y).
false.

?-gosta3(X,Y).
ERROR: gosta3/2: Undefined procedure: documentario/1
```

15. **(1.0)** Considere que está a implementar uma variante do 8-puzzle que para além de permitir jogadas horizontais e verticais, permite também fazer movimentos de peças na diagonal. Indique as linhas em PROLOG necessárias para que o predicado `mov_legal(C1,M,P,C2)` passe a considerar o movimento `ce`, que corresponde a mover uma peça na diagonal cima-esquerda, como válido. Por exemplo:

```
?-mov_legal([0,2,3,4,5,6,7,8,1], ce, 5, [5,2,3,4,0,6,7,8,1]).
true.
```

Observação: não é necessário implementar o predicado `mov_legal` todo, basta implementar as regras correspondentes ao movimento `ce`.

Resposta:

```
mov_legal([0,P2,P3,P4,P5,P6,P7,P8,P9],ce,P5,[P5,P2,P3,P4,0,P6,P7,P8,P9]).
mov_legal([P1,0,P3,P4,P5,P6,P7,P8,P9],ce,P6,[P1,P6,P3,P4,P5,0,P7,P8,P9]).
mov_legal([P1,P2,P3,0,P5,P6,P7,P8,P9],ce,P8,[P1,P2,P3,P8,P5,P6,P7,0,P9]).
mov_legal([P1,P2,P3,P4,0,P6,P7,P8,P9],ce,P9,[P1,P2,P3,P4,P9,P6,P7,P8,0]).
```