



1. (1.5) Para cada uma das seguintes questões, indique se é verdadeira ou falsa. Cada resposta certa vale 0.5 valores e *cada resposta errada desconta 0.2 valores*.

(a) Num OBDD não podem existir caminhos com ordenações incompatíveis.

**Resposta:**

Verdadeira

(b) As ordenações para BDDs  $[P, Q, S]$  e  $[R, P, T, Q, S]$  são compatíveis.

**Resposta:**

Verdadeira

(c) Em lógica de primeira ordem, uma interpretação é uma função que tem como domínio as entidades da conceptualização e como contradomínio as entidades da linguagem.

**Resposta:**

Falsa

2. (0.5) Escolha a *única* resposta *correcta* para a seguinte questão. A resposta certa vale 0.5 valor e *uma resposta errada desconta 0.2 valores*.

No PROLOG, o predicado da unificação (=):

A. Tem sucesso apenas se os dois termos forem iguais.

B. Avalia a expressão do lado direito e unifica com a expressão do lado esquerdo.

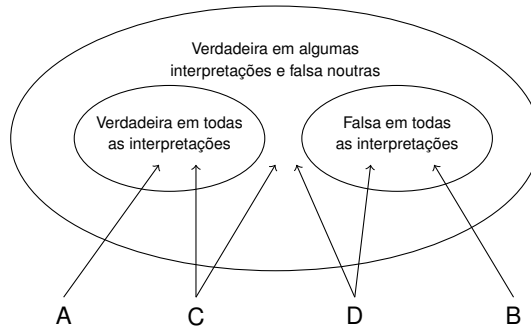
C. Avalia a expressão do lado esquerdo e unifica com a expressão do lado direito.

D. Tem sucesso se os dois termos forem unificáveis.

**Resposta:**

D

3. (1.0) Para a seguinte figura em que a oval exterior corresponde ao conjunto de todas as *fbfs*, indique quais as regiões que correspondem a *fbfs* tautológicas, contraditórias, satisfazíveis e falsificáveis. Cada resposta certa vale 0.25 valores e *cada resposta errada desconta 0.1 valores*.



**Resposta:**

A: Tautológicas

B: Contraditórias

C: Satisfazíveis

D: Falsificáveis

4. (1.0) Sendo  $\alpha_1$ ,  $\alpha_2$  e  $\alpha_3$  *fbfs* da Lógica Proposicional, escolha a única alternativa que torna *incorreta* a seguinte afirmação: " $\Delta = \{\alpha_1, \alpha_2, \alpha_3\}$  não é satisfazível se e só se ...". A resposta certa vale 1 valor e uma resposta errada desconta 0.4 valores.

A. Nenhuma das *fbfs* de  $\Delta$  é satisfazível.

B. A *fbf*  $\alpha_1 \wedge \alpha_2 \wedge \alpha_3$  não é satisfazível.

C.  $\{\alpha_1, \alpha_2\} \models \neg\alpha_3$ ."

**Resposta:**

Resposta: A

5. (1.5) Considere a conceptualização  $(D, F, R)$  em que:

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

$$F = \{\}$$

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

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

$$I(a) = \diamond$$

$$I(b) = \square$$

$$I(c) = \odot$$

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

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

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

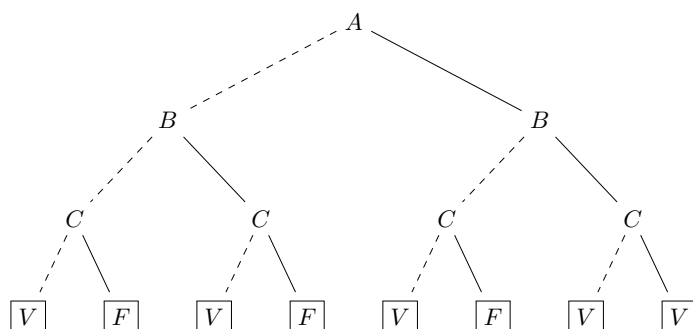
**Resposta:**

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

6. Considere a *fbf*  $(A \wedge B) \vee \neg C$ .

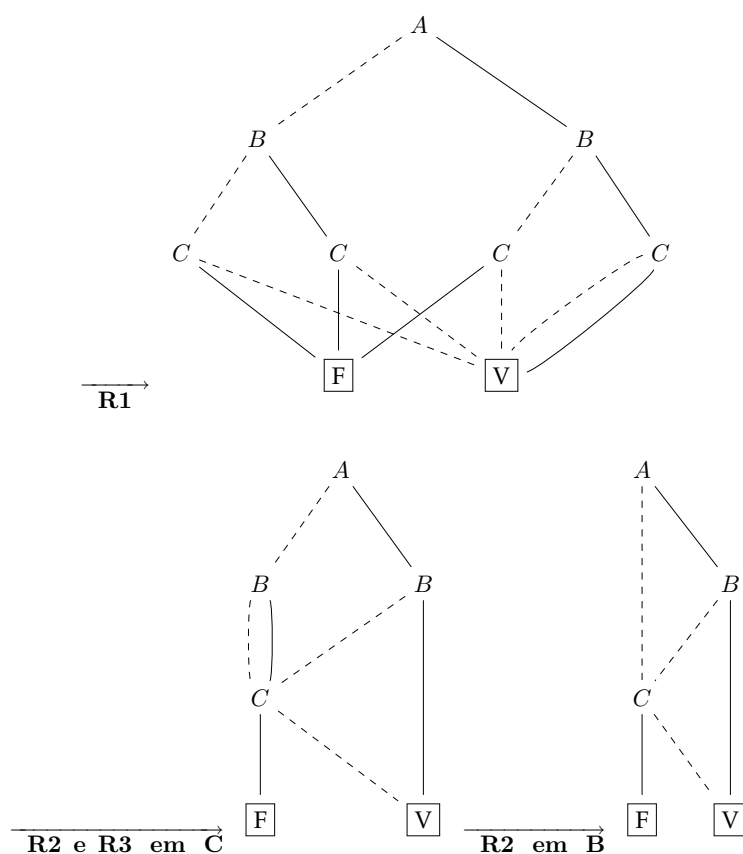
(a) (0.5) Construa a sua árvore de decisão binária.

Resposta:



(b) (1.0) Determine o seu BDD reduzido, indicando todos os passos.

Resposta:



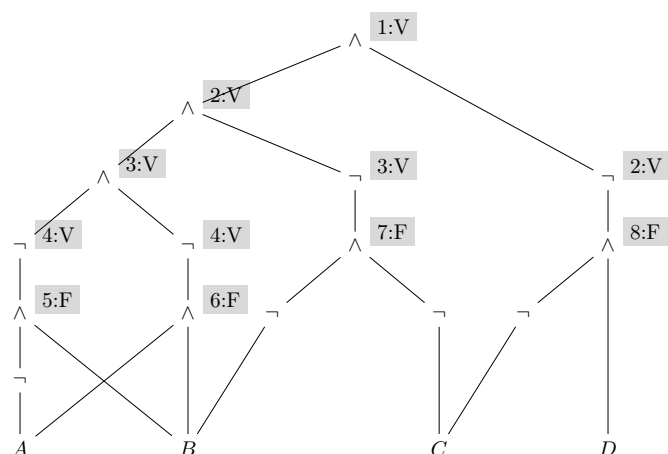
(c) (1.0) Com base no BDD reduzido, indique os seus modelos.

Resposta:

Os seus modelos correspondem aos caminhos desde a raiz até à folha  $\boxed{V}$ :

- $I(A) = I(B) = V$  e  $I(C)$  qualquer;
- $I(A) = I(C) = F$  e  $I(B)$  qualquer;
- $I(A) = V$  e  $I(C) = I(B) = F$ .

7. Considere o seguinte DAG ao qual foi aplicado o algoritmo de propagação de marcas.



- (a) (1.0) Introduza na tabela que se segue as marcas propagadas como resultado da aplicação do algoritmo de teste de nós. Por exemplo, a primeira linha representa o cenário em que a marca F é temporariamente atribuída ao nó com rótulo C. (Se existirem nós sem marcas coloque 'X' na posição respectiva.)

A	B	C	D
		F	
V			
F			

**Resposta:**

A	B	C	D
V/F	V	F	F
V	F	V	X
F	F	V	X

- (b) (0.5) Analisando o conteúdo da primeira linha da tabela anterior, o que pode concluir quanto ao nó com rótulo C? Justifique.

**Resposta:**

Como é encontrada uma contradição após a atribuição da marca F ao nó com rótulo C, podemos concluir que o nó com rótulo C tem que ser marcado com V.

- (c) (0.5) Analisando o conteúdo das duas últimas linhas da tabela anterior, o que pode concluir quanto aos nós com rótulos B e C? Justifique.

**Resposta:**

Comparando as marcas obtidas nos dois testes do nó com rótulo A, podemos passar a permanentes as marcas temporárias comuns aos dois testes, ou seja, a marca F para o nó com rótulo B e a marca V para o nó com rótulo C.

8. Considere o conjunto de cláusulas  $\Delta = \{\{P, Q, \neg R\}, \{\neg P\}, \{Q, \neg R\}, \{\neg Q, \neg R\}\}$  e o algoritmo de Davis-Putnam (DP).

- (a) (1.5) Introduza a informação em falta resultante da aplicação do algoritmo DP.

$$\exists P(\Delta) =$$

$$\exists Q(\exists P(\Delta)) =$$

$$\exists R(\exists Q(\exists P(\Delta))) =$$

**Resposta:**

$$\exists P(\Delta) = \{\{Q, \neg R\}, \{\neg Q, \neg R\}\}$$

$$\exists Q(\exists P(\Delta)) = \{\{\neg R\}\}$$

$$\exists R(\exists Q(\exists P(\Delta))) = \{\}$$

- (b) (1.0) O que pode concluir em relação à satisfazibilidade de  $\Delta$  após a aplicação do algoritmo DP? Justifique.

**Resposta:**

O conjunto de cláusulas  $\Delta$  é satisfazível porque obtemos um conjunto vazio de cláusulas como resultado da aplicação do algoritmo DP.

9. Complete a implementação dos seguintes predicados:

- (a) (0.5) `listaMenor(L1, L2)` que tem o valor verdadeiro quando o número de elementos da lista `L1` é estritamente inferior ao número de elementos da lista `L2`.

`/* Escreva aqui a condição de paragem em falta */`

`listaMenor([_|T1],[_|T2]) :- listaMenor(T1,T2).`

**Resposta:**

`listaMenor([],[_|_]).`

`listaMenor([_|T1],[_|T2]) :- listaMenor(T1,T2).`

- (b) (0.5) `ordenada(L)` que tem o valor verdadeiro quando os elementos da lista `L` estão ordenados por ordem crescente, assumido que os elementos da lista são números.

`ordenada([]).`

`ordenada([_]).`

`/* Escreva aqui a cláusula em falta */`

**Resposta:**

`ordenada([]).`

`ordenada([_]).`

`ordenada([X,Y|R]) :- X <= Y, ordenada([Y|R]).`

---

10. (1.0) Considere o seguinte programa em PROLOG:

```
xpto_aux([], L, L).  
xpto_aux([P|R], L1, [P|L2]) :- xpto_aux(R, L1, L2).  
  
xpto([], []).  
xpto([P|R], I) :- xpto(R, I1), xpto_aux(I1, [P], I).
```

Escolha a única resposta correcta para as seguintes questões. Cada resposta certa vale 0.5 valores e *cada resposta errada desconta 0.2 valores*.

(a) A resposta ao objectivo ?- `xpto([4, 5, 6], [2, 5, 6])` . é:

- A. `false`.
- B. `true`.
- C. Um ciclo infinito.
- D. Nenhuma das anteriores.

**Resposta:**

A

(b) A resposta ao objectivo ?- `xpto([2, 3, 4], [X, Y, 2])` . é:

- A. `X = [], Y=[3, 4]`.
- B. `Y= 3, X= 4`.
- C. `X = 4, Y= 3`.
- D. Nenhuma das anteriores.

**Resposta:**

C

11. (1.5) Explique por palavras o funcionamento do predicado `xpto` da pergunta anterior.

**Resposta:**

`xpto(A, B)` afirma que B é a lista que resulta de inverter a lista A.

12. (1.0) Considere a seguinte expressão em PROLOG: `V is Exp`. Qual a semântica desta expressão? Qual a razão por que a utilização da avaliação "estraga" a propriedade dos predicados em PROLOG poderem ser utilizados com qualquer argumento como dado ou como resultado?

**Resposta:**

Ao avaliar um literal da forma `V is Exp`, se a expressão `Exp` é avaliada sem erros, produzindo um valor, então se este valor é unificável com `V` a avaliação tem sucesso devolvendo a substituição adequada; em caso contrário, a avaliação falha.

Uma vez que numa avaliação se avalia o termo antes da ligação do seu valor à variável, as variáveis que eventualmente existam no termo devem estar instanciadas no momento da sua avaliação. Isto significa que com a introdução da avaliação, perdemos a possibilidade de utilizar qualquer dos argumentos como variável.

13. Considere o seguinte programa em PROLOG.

```
interruptorDesligado1(X) :- desligado(X), interruptor(X).  
interruptorDesligado2(X) :- interruptor(X), desligado(X).  
desligado(X) :- not(ligado(X)).
```

```

interruptor(i1).
interruptor(i2).
ligado(i1).

```

- (a) (0.5) Indique qual o resultado de avaliar `interruptorDesligado1(X)`.

**Resposta:**

`false.`

- (b) (0.5) Indique qual o resultado de avaliar `interruptorDesligado2(X)`.

**Resposta:**

`X = i2.`

14. Considere o projecto que implementou este ano em LP (na figura pode relembrar como é identificada cada posição).

(top, left)	(top, middle)	(top, right)
(center, left)	(center, middle)	(center, right)
(bottom, left)	(bottom, middle)	(bottom, right)

- (a) (1.0) Imagine a pista `top/2` que recebe uma peça e um tabuleiro e indica que esta pode ser colocada nas posições do topo do tabuleiro (isto é, posição 1, 2 e 3 da lista que representa o tabuleiro). Supondo que o predicado `coloca` está implementado como no projecto, uma implementação possível desta pista seria: (Escolha a única resposta correcta. Uma resposta errada desconta 0.4 valores.)

A. `top(Peca, Tabuleiro) :-  
coloca(Peca, Linha, Coluna, Tabuleiro).`

B. `top(Peca, Tabuleiro) :-  
coloca(Peca, _, _, Tabuleiro).`

C. `top(Peca, Tabuleiro) :-  
coloca(Peca, top, _, Tabuleiro).`

D. `top(Peca, Tabuleiro) :-  
coloca(Peca, top, right, Tabuleiro).  
coloca(Peca, top, left, Tabuleiro).`

E. Nenhuma das anteriores

**Resposta:**

C

- (b) (1.0) Relembre a pista `cantoTopLeft` do projecto (figura 1) e considere o predicado `meio/2`, implementado como se segue:

```

meio(Peca, Tabuleiro) :-
coloca(Peca, center, middle, Tabuleiro).

```

cantoTopLeft

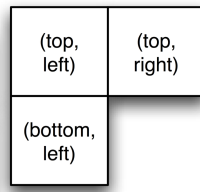


Figure 1: Pista cantoTopLeft.

Uma implementação possível desta pista seria: (Escolha a única resposta correcta. Uma resposta errada desconta 0.4 valores.)

- A. `cantoTopLeft(Peca, Linha, Coluna, Tabuleiro) :-  
    (coloca(Peca, Linha, Coluna, Tabuleiro);  
    meio(Peca, Tabuleiro);  
    coloca(Peca, center, Coluna, Tabuleiro)).`
- B. `cantoTopLeft(Peca, Linha, Coluna, Tabuleiro) :-  
    (coloca(Peca, Linha, _, Tabuleiro);  
    coloca(Peca, Linha, middle, Tabuleiro);  
    meio(Peca, Tabuleiro);  
    coloca(Peca, center, _, Tabuleiro)).`
- C. `cantoTopLeft(Peca, Linha, Coluna, Tabuleiro) :-  
    (coloca(Peca, Linha, Coluna, Tabuleiro);  
    coloca(Peca, Linha, middle, Tabuleiro);  
    meio(Peca, Tabuleiro);  
    coloca(Peca, center, Coluna, Tabuleiro)).`
- D. `cantoTopLeft(Peca, Linha, Coluna, Tabuleiro) :-  
    (coloca(Peca, Linha, Coluna, Tabuleiro);  
    meio(Peca, Tabuleiro)).`
- E. Nenhuma das anteriores

**Resposta:**

C