Exercícios de Lógica para Programação

Ana Cardoso-Cachopo

Fevereiro de 2014

Conteúdo

1 Argumentos e Validade 3

2 Lógica Proposicional — Sistema de Dedução Natural 4

3 Lógica Proposicional — Tabelas de Verdade 6

4 Lógica Proposicional — Resolução 7

5 Lógica Proposicional — BDDs 8

6 Lógica Proposicional — OBDDs 10

7 Lógica Proposicional — SAT 11

8 Lógica de Primeira Ordem — Sistema de Dedução Natural 13

9 Lógica de Primeira Ordem — Sistema Semântico 15

10 Lógica de Primeira Ordem — Representação 19

11 Lógica de Primeira Ordem — Resolução 20

12 Programação em Lógica — Resolução SLD; Árvores SLD 22

13 Prolog — Árvores de Refutação; Listas 24

14 Prolog — Operadores Pré-definidos 26

15 Prolog — Corte; Negação 28

Prefácio

Este documento contém uma compilação de exercícios para a disciplina de Lógica para Programação da LEIC.

A maior parte dos exercícios foi criada por mim especificamente para as aulas práticas ou para as provas de avaliação da disciplina e outros foram tirados de livros ou artigos acerca da matéria em questão.

Existem dois exercícios cujo enunciado foi feito pelo Professor João Pavão Martins nas aulas sobre os sistemas de dedução natural da lógica proposicional e da lógica de primeira ordem. Esses exercícios estão assinalados com a etiqueta (JPM).

O Professor João Cachopo, para além de discutir comigo algumas das respostas dos exercícios, ajudou-me a fazer em LaTeX as figuras dos vários capítulos, tornando esta compilação (muito) mais apresentável.

Obviamente, a responsabilidade por quaisquer erros ou gralhas que esta compilação de exercícios possa ter é inteiramente minha.

Correcções de gralhas ou sugestões de melhorias podem ser enviadas para o meu endereço de email: `acardoso@tecnico.ulisboa.pt`.

1 Argumentos e Validade

Exercício 1.1
Usando apenas a informação que está explícita, diga, justificando, se os seguintes argumentos são válidos ou são inválidos:

1. Peregrino Cinzento é Gandalf
   Mithrandir é Gandalf
   ∴ Peregrino Cinzento é Mithrandir
2. Mithrandir é um feiticeiro
   Mithrandir é Gandalf
   ∴ Gandalf é um feiticeiro
3. Os orcs são feios
   ∴ Os orcs são feios
4. Nemo é um peixe
   Dori é um peixe
   ∴ Nemo é Dori
5. Os tubarões são carnívoros
   Os tubarões não são vegetarianos
   O Bruce é vegetariano
   ∴ O Bruce não é tubarão
6. Os peixes são animais
   ∴ Os tubarões são animais

Exercício 1.2
Sempre que for possível, dê exemplos de argumentos válidos e inválidos com:

- As premissas verdadeiras e a conclusão verdadeira;
- As premissas verdadeiras e a conclusão falsa;
- As premissas falsas e a conclusão verdadeira;
- As premissas falsas e a conclusão falsa.

2 Lógica Proposicional — Sistema de Dedução Natural

Exercício 2.1
(JPM) Demonstre os seguintes teoremas e argumentos usando o sistema de dedução natural da lógica proposicional. Em cada alínea indique se está a demonstrar um teorema ou um argumento.

1. $A \to (B \to A)$
2. $(A \land \neg A) \to B$
3. $(\{A \to B, B \to \neg A\}, \neg A)$
4. $(\{A\}, B \to (A \land B))$
5. $(\{\}, ((A \to B) \land (B \to C)) \to (A \to C))$

Exercício 2.2
Demonstre os seguintes teoremas usando o sistema de dedução natural da lógica proposicional.

1. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
2. $((A \to (A \to B)) \land A) \to B$
3. $(A \to B) \to (\neg B \to \neg A)$
4. $(\neg B \to \neg A) \to (A \to B)$
5. $(A \land B) \to \neg(A \to \neg B)$
6. $((A \to B) \land \neg B) \to \neg A$
7. $(A \to \neg A) \to \neg A$
8. $(A \lor B) \to (B \lor A)$
9. $((A \lor B) \lor C) \to (A \lor (B \lor C))$
10. $(A \land (B \lor C)) \to ((A \land B) \lor (A \land C))$
11. $((A \land B) \lor (A \land C)) \to (A \land (B \lor C))$
12. $((A \land B) \lor A) \to A$

Exercício 2.3
Demonstre os seguintes teoremas, que correspondem a aplicações das leis de De Morgan, usando o sistema de dedução natural da lógica proposicional.

1. $\neg(A \lor B) \to (\neg A \land \neg B)$
2. $(\neg A \land \neg B) \to \neg(A \lor B)$
3. $\neg(A \land B) \to (\neg A \lor \neg B)$
4. $(\neg A \lor \neg B) \to \neg(A \land B)$

Lógica Proposicional — Sistema de Dedução Natural

Prem

$$\begin{array}{lll} n & \alpha & \text{Prem} \end{array}$$

Hip

$$\begin{array}{l|ll} n & \alpha & \text{Hip} \\ \cline{2-2} n+1 & \ldots & \end{array}$$

Rep

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \alpha & \text{Rep}, n \end{array}$$

Rei

$$\begin{array}{llll} n & \alpha & & \\ \vdots & & \multicolumn{1}{|l}{\vdots} & \\ m & & \multicolumn{1}{|l}{\alpha} & \text{Rei}, n \end{array}$$

I $\rightarrow$

$$\begin{array}{l|ll} n & \alpha & \text{Hip} \\ \cline{2-2} \vdots & \vdots & \\ m & \beta & \\ \multicolumn{1}{l}{k} & \alpha \rightarrow \beta & \text{I}\rightarrow, (n, m) \end{array}$$

E $\rightarrow$

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \alpha \rightarrow \beta & \\ \vdots & \vdots & \\ k & \beta & \text{E}\rightarrow, (n, m) \end{array}$$

I$\wedge$

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \beta & \\ \vdots & \vdots & \\ k & \alpha \wedge \beta & \text{I}\wedge, (n, m) \end{array}$$

E$\wedge$

$$\begin{array}{lll} n & \alpha \wedge \beta & \\ \vdots & \vdots & \\ m & \alpha & \text{E}\wedge, n \end{array}$$

ou

$$\begin{array}{lll} n & \alpha \wedge \beta & \\ \vdots & \vdots & \\ m & \beta & \text{E}\wedge, n \end{array}$$

I$\vee$

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \alpha \vee \beta & \text{I}\vee, n \end{array}$$

ou

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \beta \vee \alpha & \text{I}\vee, n \end{array}$$

E$\vee$

$$\begin{array}{llll} n & \alpha \vee \beta & & \\ o & & \multicolumn{1}{|l}{\alpha} & \text{Hip} \\ \cline{3-3} \vdots & & \multicolumn{1}{|l}{\vdots} & \\ p & & \multicolumn{1}{|l}{\gamma} & \\ r & & \multicolumn{1}{|l}{\beta} & \text{Hip} \\ \cline{3-3} \vdots & & \multicolumn{1}{|l}{\vdots} & \\ s & & \multicolumn{1}{|l}{\gamma} & \\ m & \gamma & & \text{E}\vee, (n, (o, p), (r, s)) \end{array}$$

I$\neg$

$$\begin{array}{l|ll} n & \alpha & \text{Hip} \\ \cline{2-2} \vdots & \vdots & \\ m & \beta & \\ \vdots & \vdots & \\ k & \neg\beta & \\ \multicolumn{1}{l}{l} & \neg\alpha & \text{I}\neg, (n, (m, k)) \end{array}$$

E$\neg$

$$\begin{array}{lll} n & \neg\neg\alpha & \\ \vdots & \vdots & \\ m & \alpha & \text{E}\neg, n \end{array}$$

3 Lógica Proposicional — Tabelas de Verdade

Exercício 3.1
Para cada uma das seguintes fbfs, diga, justificando, se é satisfazível, falsificável, tautológica ou contraditória. É necessário conhecer todas as linhas das tabelas de verdade para responder a estas perguntas?

1. A
2. $A \wedge \neg A$
3. $A \vee \neg A$
4. $(A \wedge B) \rightarrow (A \vee B)$
5. $(A \vee B) \rightarrow (A \wedge B)$

Exercício 3.2
Considere o seguinte conjunto de fórmulas:

$$\{Homem \rightarrow Pessoa, Mulher \rightarrow Pessoa, Homem \vee Mulher\}$$

1. Mostre quais são os modelos desse conjunto.
2. $Pessoa$ é consequência lógica desse conjunto? Porquê?
3. Acrescente $\neg Homem$ ao conjunto. Diga quais são os seus modelos e as suas consequências lógicas.

Exercício 3.3
Considere a seguinte tabela de verdade. Diga a que conectiva lógica corresponde a função f.

P	Q	$f(P,Q)$
V	V	V
V	F	F
F	V	F
F	F	F

Exercício 3.4
Usando o sistema semântico da lógica proposicional, mostre que $\{\neg A, \neg B\} \models \neg(A \vee B)$.

Exercício 3.5
Mostre que as fórmulas $A \rightarrow (B \rightarrow C)$ e $(A \rightarrow B) \rightarrow C$ não são equivalentes apresentando uma interpretação para a qual elas tenham valores lógicos diferentes.

Exercício 3.6
Usando tabelas de verdade, prove que as seguintes fórmulas, correspondentes às leis de De Morgan, são tautologias.

$$\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$$

$$\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$$

4 Lógica Proposicional — Resolução

Exercício 4.1
Transforme a seguinte fórmula para a forma clausal:

1. $A \rightarrow (\neg(\neg B \vee C) \vee (C \rightarrow D))$

Exercício 4.2
Demonstre os teoremas e argumentos do exercício 2.1 usando resolução. Em cada alínea indique se está a demonstrar um teorema ou um argumento.

Exercício 4.3
Demonstre os teoremas do exercício 2.2 usando resolução.

Exercício 4.4
Demonstre os teoremas do exercício 2.3, que correspondem a aplicações das leis de De Morgan, usando resolução.

Exercício 4.5
Usando uma estratégia de resolução linear, apresente uma prova por refutação para A a partir do seguinte conjunto de cláusulas: $\{\{A, B, \neg C\}, \{\neg D, A\}, \{\neg B\}, \{C\}\}$.

5 Lógica Proposicional — BDDs

Exercício 5.1
Considere a seguinte tabela de verdade:

P	Q	$f(P,Q)$
V	V	V
V	F	F
F	V	F
F	F	F

1. Represente a árvore de decisão correspondente.
2. Mostre o BDD reduzido correspondente à árvore anterior, apresentando e justificando todos os passos.
3. Com base no BDD reduzido, diga quais são os modelos de $f(P,Q)$ e compare-os com os obtidos pela observação da tabela de verdade.

Exercício 5.2
Considere a seguinte tabela de verdade:

P	Q	R	$f(P,Q,R)$
V	V	V	V
V	V	F	F
V	F	V	F
V	F	F	F
F	V	V	V
F	V	F	F
F	F	V	V
F	F	F	F

1. Represente a árvore de decisão correspondente.
2. Mostre o BDD reduzido correspondente à árvore anterior, apresentando e justificando todos os passos.
3. O BDD teria a mesma forma se tivesse escolhido outra ordenação para os nós? Justifique, mostrando o novo BDD reduzido.
4. Com base nos BDDs reduzidos, diga quais são os modelos de $f(P,Q,R)$ e compare-os com os obtidos pela observação da tabela de verdade.

Exercício 5.3
Usando BDDs, prove que as seguintes fórmulas, correspondentes às leis de De Morgan, são tautologias.

$$\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$$
$$\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$$

Exercício 5.4
Considere que $\alpha = A \wedge B \wedge C$ e $\beta = B \vee C$. Determine os BDDs reduzidos para:

1. α
2. β
3. $\alpha \wedge \beta$
4. $\alpha \vee \neg\beta$

6 Lógica Proposicional — OBDDs

Exercício 6.1
Considere as seguintes fbfs:

1. $\alpha = A \wedge B \wedge C$
2. $\beta = B \vee C$
3. $\gamma = (A \wedge B) \vee D$

Partindo das suas árvores de decisão binárias, mostre os seus OBBDs reduzidos através da aplicação do algoritmo *reduz*. Deve usar a ordenação $[A, B, C, D]$ para os predicados.

Exercício 6.2
Considerando os OBBDs reduzidos do exercício anterior, combine-os usando o algoritmo *aplica* (se isso for possível) para obter os OBBDs reduzidos para as seguintes fbfs.

1. $\alpha \wedge \beta$
2. $\alpha \vee \neg\beta$
3. $\beta \vee \gamma$

7 Lógica Proposicional — SAT

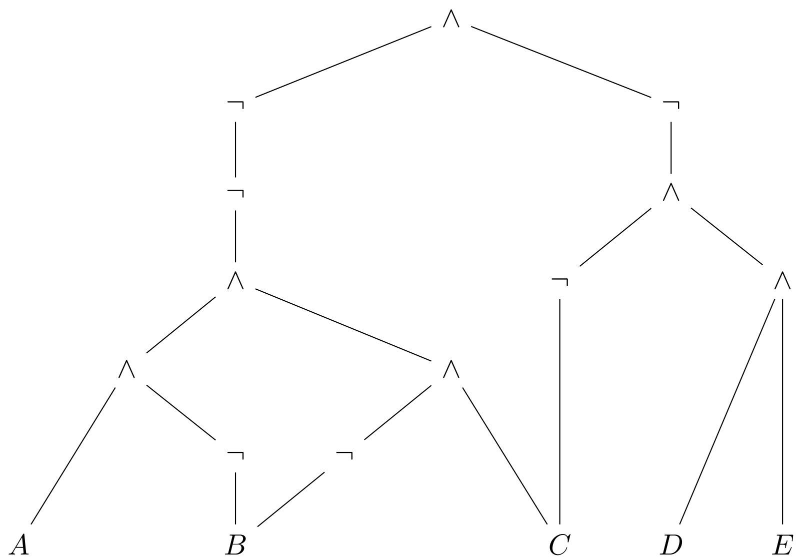
Exercício 7.1
Considere a seguinte fórmula: $\neg(A \to (B \vee C)) \wedge ((B \vee C) \vee D)$. Crie o seu DAG, efectue a propagação de valores de modo a que a fórmula seja verdadeira e apresente uma testemunha.

Exercício 7.2
Considere a seguinte fórmula: $(A \wedge \neg B) \wedge (B \wedge \neg(B \wedge C))$.

1. Crie o seu DAG. Efectue a propagação de valores, de modo a que a fórmula seja verdadeira. O que pode concluir?

2. Faça o mesmo para a negação da fórmula dada. Consegue encontrar uma testemunha usando apenas o algoritmo de propagação de marcas? E usando o algoritmo de teste de nós? O que pode concluir?

Exercício 7.3
Considere o seguinte DAG. Diga a que fórmula corresponde. Efectue a propagação de valores, de modo a que a fórmula seja verdadeira. Se for necessário, use o algoritmo de teste de nós. Se for possível, apresente uma testemunha.



Exercício 7.4
Considere o seguinte conjunto de cláusulas: $\{\{\neg A, \neg B\}, \{A, \neg B\}, \{B\}\}$. Aplique o algoritmo DP e caso a fórmula seja satisfazível, indique uma testemunha.

1. Usando a ordem $A \prec B$;

2. Usando a ordem $B \prec A$.

Exercício 7.5
Considere o seguinte conjunto de cláusulas: $\{\{\neg D, B\}, \{\neg C, A\}, \{\neg A, D, C\}, \{\neg C, E\}, \{\neg E\}\}$. Aplique o algoritmo DP usando a ordem $C \prec E \prec D \prec A \prec B$. Caso a fórmula seja satisfazível, indique uma testemunha.

Exercício 7.6
Considere a seguinte fórmula $((A \to B) \vee (A \to C)) \wedge (\neg A \vee D)$.
Determine se ela é satisfazível e se for indique uma testemunha, usando:

1. o algoritmo de propagação de marcas (e o algoritmo de teste de nós, se necessário);
2. o algoritmo DP.

8 Lógica de Primeira Ordem — Sistema de Dedução Natural

Exercício 8.1
Demonstre os seguintes argumentos, usando o sistema de dedução natural da lógica de primeira ordem:

1. $(\{\forall x[F(x)]\}, \exists x[F(x)])$
2. $(\{\forall x[F(x) \rightarrow G(x)], \exists x[F(x) \wedge H(x)]\}, \exists x[G(x) \wedge H(x)])$

Exercício 8.2
(JPM) Demonstre os seguintes teoremas, usando o sistema de dedução natural da lógica de primeira ordem:

1. $(F(a) \wedge \forall x[F(x) \rightarrow G(x)]) \rightarrow G(a)$
2. $(\forall x[F(x) \rightarrow G(x)] \wedge \forall x[G(x) \rightarrow H(x)]) \rightarrow \forall x[F(x) \rightarrow H(x)]$
3. $(\forall x[F(x) \rightarrow H(x)] \wedge \exists y[F(y)]) \rightarrow \exists z[H(z)]$

Exercício 8.3
Demonstre os seguintes teoremas, que correspondem a aplicações das leis de De Morgan para os quantificadores, usando o sistema de dedução natural da lógica de primeira ordem.

1. $\neg\forall x[F(x)] \rightarrow \exists x[\neg F(x)]$
2. $\exists x[\neg F(x)] \rightarrow \neg\forall x[F(x)]$
3. $\neg\exists x[F(x)] \rightarrow \forall x[\neg F(x)]$
4. $\forall x[\neg F(x)] \rightarrow \neg\exists x[F(x)]$

Lógica de Primeira Ordem — Sistema de Dedução Natural

Prem

$$\begin{array}{lll} n & \alpha & \text{Prem} \end{array}$$

Hip

$$\begin{array}{l|ll} n & \alpha & \text{Hip} \\ \cline{2-2} n+1 & \ldots & \end{array}$$

Rep

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \alpha & \text{Rep}, n \end{array}$$

Rei

$$\begin{array}{ll|ll} n & \alpha & & \\ \vdots & & \vdots & \\ m & & \alpha & \text{Rei}, n \end{array}$$

I →

$$\begin{array}{l|ll} n & \alpha & \text{Hip} \\ \cline{2-2} \vdots & \vdots & \\ m & \beta & \\ k & \multicolumn{1}{l}{\alpha \to \beta} & \text{I}\to, (n, m) \end{array}$$

E →

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \alpha \to \beta & \\ \vdots & \vdots & \\ k & \beta & \text{E}\to, (n, m) \end{array}$$

I∧

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \beta & \\ \vdots & \vdots & \\ k & \alpha \land \beta & \text{I}\land, (n, m) \end{array}$$

E∧

$$\begin{array}{lll} n & \alpha \land \beta & \\ \vdots & \vdots & \\ m & \alpha & \text{E}\land, n \end{array}$$

ou

$$\begin{array}{lll} n & \alpha \land \beta & \\ \vdots & \vdots & \\ m & \beta & \text{E}\land, n \end{array}$$

I∨

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \alpha \lor \beta & \text{I}\lor, n \end{array}$$

ou

$$\begin{array}{lll} n & \alpha & \\ \vdots & \vdots & \\ m & \beta \lor \alpha & \text{I}\lor, n \end{array}$$

E∨

$$\begin{array}{l|ll} n & \multicolumn{1}{l}{\alpha \lor \beta} & \\ o & \alpha & \text{Hip} \\ \cline{2-2} \vdots & \vdots & \\ p & \gamma & \\ r & \beta & \text{Hip} \\ \cline{2-2} \vdots & \vdots & \\ s & \gamma & \\ m & \multicolumn{1}{l}{\gamma} & \text{E}\lor, (n, (o, p), (r, s)) \end{array}$$

I¬

$$\begin{array}{l|ll} n & \alpha & \text{Hip} \\ \cline{2-2} \vdots & \vdots & \\ m & \beta & \\ \vdots & \vdots & \\ k & \neg\beta & \\ l & \multicolumn{1}{l}{\neg\alpha} & \text{I}\neg, (n, (m, k)) \end{array}$$

E¬

$$\begin{array}{lll} n & \neg\neg\alpha & \\ \vdots & \vdots & \\ m & \alpha & \text{E}\neg, n \end{array}$$

I∀

$$\begin{array}{ll|ll} n & x_0 & & \\ \vdots & & \vdots & \\ m & & \alpha(x_0) & \\ m+1 & \multicolumn{2}{l}{\forall x[\alpha(x)]} & \text{I}\forall, (n, m) \end{array}$$

E∀

$$\begin{array}{lll} n & \forall x[\alpha(x)] & \\ \vdots & \vdots & \\ m & \alpha(t) & \text{E}\forall, n \end{array}$$

I∃

$$\begin{array}{lll} n & \alpha(t) & \\ \vdots & \vdots & \\ m & \exists x[\alpha(x)] & \text{I}\exists, n \end{array}$$

E∃

$$\begin{array}{ll|ll} n & \multicolumn{2}{l}{\exists x[\alpha(x)]} & \\ m & x_0 & \alpha(x_0) & \text{Hip} \\ \cline{3-3} \vdots & & \vdots & \\ k & & \beta & \\ k+1 & \multicolumn{2}{l}{\beta} & \text{E}\exists, (n, (m, k)) \end{array}$$

9 Lógica de Primeira Ordem — Sistema Semântico

Exercício 9.1
Considere a seguinte conceptualização $C = (D, F, R)$:

- $D = \{✦, ✶, ✷, ✢, ✻, ❋\}$
- $F = \{\{(✦, ✢), (✶, ✻), (✷, ❋)\}\}$
- $R = \{\{(✦), (✶), (✷)\}, \{(✢), (✻), (❋)\}, \{(✦, ✶), (✦, ✷), (✶, ✷)\}\}$

e a seguinte interpretação:

- $I(eq) = ✦$
- $I(es) = ✶$
- $I(eo) = ✷$
- $I(fq) = ✢$
- $I(fs) = ✻$
- $I(fo) = ❋$
- $I(florDe) = \{(✦, ✢), (✶, ✻), (✷, ❋)\}$
- $I(Estrela) = \{(✦), (✶), (✷)\}$
- $I(Flor) = \{(✢), (✻), (❋)\}$
- $I(MenosPontas) = \{(✦, ✶), (✦, ✷), (✶, ✷)\}$

Diga, justificando, quais das seguintes fbfs são satisfeitas por esta interpretação para esta conceptualização:

1. $Estrela(eo)$
2. $Estrela(florDe(eq))$
3. $MenosPontas(eo, eq)$
4. $\neg Estrela(es) \vee Flor(fq)$
5. $\exists x[Estrela(x)]$
6. $\forall x, y, z[(MenosPontas(x, y) \wedge MenosPontas(y, z)) \rightarrow MenosPontas(x, z)]$

Exercício 9.2
Considere a seguinte conceptualização $C = (D, F, R)$:

- $D = \{I, V, X, C, D, M, 1, 5, 10, 100, 500, 1000\}$
- $F = \{\{(I, 1), (V, 5), (X, 10), (C, 100), (D, 500), (M, 1000)\}\}$
- $R = \{\{(X), (C), (D), (M)\}, \{(I), (V)\}\}$

e a seguinte interpretação:

- $I(um) \mapsto I$
- $I(cinco) \mapsto V$

- $I(dez) \mapsto X$
- $I(cem) \mapsto C$
- $I(quinhentos) \mapsto D$
- $I(mil) \mapsto M$
- $I(valor) \mapsto \{(I, 1), (V, 5), (X, 10), (C, 100), (D, 500), (M, 1000)\}$
- $I(Par) \mapsto \{(X), (C), (D), (M)\}$
- $I(Impar) \mapsto \{(I), (V)\}$

1. Diga, justificando, quais das seguintes fbfs são satisfeitas por esta interpretação para esta conceptualização:
   (a) $Par(cem)$
   (b) $Impar(valor(cinco))$
   (c) $Par(cem) \rightarrow Impar(dez)$
   (d) $\forall x[Par(x)]$
   (e) $\exists x[Impar(valor(x))]$
2. Diga que alterações faria na conceptualização para os resultados serem os intuitivamente esperados. Que mudanças é que isso implicaria na interpretação?
3. Diga, justificando, se o seguinte poderia ser uma interpretação para esta conceptualização. Se não puder, indique todas as razões para não poder.
   - $I(a) \mapsto X$
   - $I(a) \mapsto 10$
   - $I(b) \mapsto 100$
   - $I(c) \mapsto 100$
   - $I(d) \mapsto 20$
   - $I(f1) \mapsto \{(I, 1), (V, 5), (V, 10), (C, 100)\}$
   - $I(R1) \mapsto \{(X), (C), (D)\}$

Exercício 9.3
Considere a relação binária $EDivisorDe$ no domínio $\{1, 2, 3, 4, 5, 6\}$.

1. Apresente o conjunto de todos os pares que estão na relação para o domínio dado.
2. Diga, justificando, se a relação é: reflexiva, simétrica, ou transitiva.

Exercício 9.4
Considere a seguinte fbf: $\forall x, y[(A(x) \wedge B(x, y)) \rightarrow A(y)]$. Diga se ela é verdadeira ou falsa para cada uma das seguintes interpretações. Se for falsa, apresente um contra-exemplo.

1. O domínio dos números naturais, onde $A(x)$ é interpretado como "x é par" e $B(x, y)$ é interpretado como "x é igual a y".
2. O domínio dos números naturais, onde $A(x)$ é interpretado como "x é par" e $B(x, y)$ é interpretado como "x é um inteiro divisor de y".
3. O domínio dos números naturais, onde $A(x)$ é interpretado como "x é par" e $B(x, y)$ é interpretado como "x é um inteiro múltiplo de y".

4. O domínio dos booleanos $\{verdadeiro, falso\}$, onde $A(x)$ é interpretado como "x é falso" e $B(x, y)$ é interpretado como "x é igual a y".

Exercício 9.5
Considere a seguinte conceptualização:

- Universo de discurso = {✿, ❀}
- Conjunto de funções = {{(✿, ❀)}, {(✿, ✿), (❀, ✿)}}
- Conjunto de relações = {{(✿, ✿), (✿, ❀), (❀, ❀)}}

e o seguinte conjunto de fórmulas: $\{P_1(f_1(a), b), P_1(f_2(a), b) \to P_1(b, f_1(a))\}$.

Diga se a seguinte interpretação é modelo deste conjunto de fórmulas:

- $I(a) \mapsto$ ✿
- $I(b) \mapsto$ ❀
- $I(f_1) \mapsto$ {(✿, ❀)}
- $I(f_2) \mapsto$ {(✿, ✿), (❀, ✿)}
- $I(P_1) \mapsto$ {(✿, ✿), (✿, ❀), (❀, ❀)}

Exercício 9.6
Considere a seguinte conceptualização $C = (D, F, R)$:

- $D = \{\boxplus, \boxminus, \boxtimes, \boxdot\}$
- $F = \{\{(\boxplus, \boxtimes)\}, \{(\boxminus, \boxdot)\}, \{(\boxminus, \boxdot, \boxtimes), (\boxminus, \boxtimes, \boxdot)\}\}$
- $R = \{\{(\boxplus, \boxdot)\}, \{(\boxtimes, \boxdot)\}, \{(\boxplus), (\boxminus), (\boxtimes), (\boxdot)\}\}$

e a seguinte interpretação:

- $I(a) = \boxplus$
- $I(b) = \boxminus$
- $I(c) = \boxtimes$
- $I(d) = \boxdot$
- $I(f1) = \{(\boxplus, \boxtimes)\}$
- $I(f2) = \{(\boxminus, \boxdot)\}$
- $I(R1) = \{(\boxplus, \boxdot)\}$
- $I(R2) = \{(\boxtimes, \boxdot)\}$
- $I(R3) = \{(\boxplus), (\boxminus), (\boxtimes), (\boxdot)\}$

1. Diga, justificando, se esta interpretação para esta conceptualização é um modelo do seguinte conjunto de fórmulas: $\{\exists x, y[R3(x) \wedge R1(x, f2(y))], R2(f1(a), d) \vee \neg R3(b)\}$.
2. Explique porque é que o seguinte não pode ser uma interpretação para esta conceptualização, mencionando todos os erros que foram cometidos.
   - $I(a) = \boxplus$
   - $I(b) = \boxplus$

- $I(c) = \boxplus$
- $I(c) = \boxminus$
- $I(d) = \circledast$
- $I(f1) = \{(\boxplus, \boxminus), (\boxplus, \boxtimes)\}$
- $I(R1) = \{(\boxplus), (\boxtimes), (\boxdot)\}$

Exercício 9.7
Represente em lógica de primeira ordem cada uma das seguintes frases:

1. O Miau é um gato castanho.
2. Os gatos são animais.
3. Nenhum gato é um cão.
4. Nem todos os gatos gostam de leite.
5. Os gatos não gostam de cães.
   (Nenhum gato gosta de nenhum cão.)
6. Existe um cão de quem todos os gatos gostam.
   (Todos os gatos gostam do mesmo cão.)
7. Todos os gatos gostam de algum cão.
   (Pode ser um cão diferente para cada gato.)
8. Se algum gato gostar do Rui, então o Miau também gosta do Rui.
9. O Miau é um siamês ou um bobtail, mas não os dois simultaneamente.
   (Não pode usar o ou exclusivo.)
10. A cauda do Miau é comprida.
11. Todos os gatos têm cauda.

Exercício 9.8
Suponha que $N(x)$ representa o predicado "x é um número", $P(x)$ representa o predicado "x é par", $I(x)$ representa o predicado "x é ímpar" e $M(x, y)$ representa o predicado "x é maior que y". Traduza as seguintes fbfs para linguagem comum. Se não conseguir traduzir alguma delas, explique porquê.

1. $\exists x[N(x)]$
2. $\forall x[I(x) \to N(x)]$
3. $\neg\forall x[N(x) \to P(x)]$
4. $\exists x[N(x) \land \neg P(x)]$
5. $\forall x[I(x) \to \neg P(x)]$
6. $\forall x[P(x) \to \exists y[I(y) \land M(y, x)]]$
7. $\exists x[P(x) \land \forall y[(P(y) \land x \neq y) \to M(y, x)]]$
8. $\exists x[P(x) \to N(x)]$

10 Lógica de Primeira Ordem — Representação

Exercício 10.1
Represente em lógica de primeira ordem cada uma das seguintes frases:

1. O Miau é um gato castanho.
2. Os gatos são animais.
3. Nenhum gato é um cão.
4. Nem todos os gatos gostam de leite.
5. Os gatos não gostam de cães.
   (Nenhum gato gosta de nenhum cão.)
6. Existe um cão de quem todos os gatos gostam.
   (Todos os gatos gostam do mesmo cão.)
7. Todos os gatos gostam de algum cão.
   (Pode ser um cão diferente para cada gato.)
8. Se algum gato gostar do Rui, então o Miau também gosta do Rui.
9. O Miau é um siamês ou um bobtail, mas não os dois simultaneamente.
   (Não pode usar o ou exclusivo.)
10. A cauda do Miau é comprida.
11. Todos os gatos têm cauda.

Exercício 10.2
Suponha que $N(x)$ representa o predicado "x é um número", $P(x)$ representa o predicado "x é par", $I(x)$ representa o predicado "x é ímpar" e $M(x,y)$ representa o predicado "x é maior que y". Traduza as seguintes fbfs para linguagem comum. Se não conseguir traduzir alguma delas, explique porquê.

1. $\exists x[N(x)]$
2. $\forall x[I(x) \rightarrow N(x)]$
3. $\neg\forall x[N(x) \rightarrow P(x)]$
4. $\exists x[N(x) \wedge \neg P(x)]$
5. $\forall x[I(x) \rightarrow \neg P(x)]$
6. $\forall x[P(x) \rightarrow \exists y[I(y) \wedge M(y,x)]]$
7. $\exists x[P(x) \wedge \forall y[(P(y) \wedge x \neq y) \rightarrow M(y,x)]]$
8. $\exists x[P(x) \rightarrow N(x)]$

11 Lógica de Primeira Ordem — Resolução

Exercício 11.1
Utilize o algoritmo de unificação para determinar quais dos seguintes conjuntos de fbfs são unificáveis, e, no caso de o serem, determine o unificador mais geral. Mostre todos os passos intermédios usados no cálculos.

1. $\{P(a, x, x), P(a, b, c)\}$
2. $\{P(a, x, f(x)), P(x, y, z)\}$
3. $\{P(x, y), Q(x, y)\}$
4. $\{Colocou(x_1, SenhorAneis, y_1), Colocou(Maria, x_2, topo(y2)),$
   $Colocou(x_3, SenhorAneis, topo(MesaAzul))\}$

Exercício 11.2
Passe as seguintes *fbfs* da lógica de primeira ordem para a forma clausal.

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

Exercício 11.3
Considere o seguinte conjunto de cláusulas: $\{\{\neg A(x), B(x), C(x)\}, \{A(x)\}, \{\neg B(a)\}, \{\neg A(y), \neg C(y)\}\}$.

1. Apresente uma demonstração por refutação a partir desse conjunto.
2. Apresente uma demonstração por refutação a partir desse conjunto, usando resolução unitária.
3. Apresente uma demonstração por refutação a partir desse conjunto, usando resolução linear e $\{\neg B(a)\}$ como cláusula inicial.

Exercício 11.4
Demonstre, usando resolução, o seguinte argumento:
$(\{\forall x, y, z[(R(x, y) \wedge R(y, z)) \rightarrow R(x, z)], R(a, b), R(b, c), R(c, d)\}, R(a, d))$.

Exercício 11.5
Demonstre os argumentos do exercício 8.1, usando resolução.

Exercício 11.6
Demonstre os teoremas do exercício 8.2, usando resolução.

Exercício 11.7
Demonstre os teoremas do exercício 8.3, que correspondem a aplicações das leis de De Morgan para os quantificadores, usando resolução.

Exercício 11.8
Considere as seguintes afirmações:

- Os animais com pêlos são mamíferos.
- Os ursos são animais com pêlos.
- Os coelhos são mamíferos.
- O Winnie é um urso.
- O Bugsbunny é um coelho.
- O Sylvester é um animal com pêlos.

1. Represente-as em lógica de primeira ordem.
2. Usando resolução, responda às seguintes perguntas:
   (a) O Winnie é mamífero?
   (b) Quem é que tem pêlos?
   (c) Quais são os mamíferos?

12 Programação em Lógica — Resolução SLD; Árvores SLD

Exercício 12.1
Demonstre os argumentos do exercício 8.1, usando resolução SLD e uma função de selecção que escolhe o primeiro literal da cláusula objectivo.
Notas: Como está a tentar provar a validade de argumentos, deve fazer provas por refutação em que uma das cláusulas corresponde à negação da conclusão. Uma vez que a passagem para a forma clausal já foi feita na aula sobre resolução, apresentam-se com cada argumento as cláusulas que lhe correspondem.

1. Argumento: $(\{\forall x[F(x)]\}, \exists x[F(x)])$
   Cláusulas: $\{\{F(x)\}, \{\neg F(y)\}\}$
2. Argumento: $(\{\forall x[F(x) \to G(x)], \exists x[F(x) \wedge H(x)]\}, \exists x[G(x) \wedge H(x)])$
   Cláusulas: $\{\{\neg F(x), G(x)\}, \{F(a)\}, \{H(a)\}, \{\neg G(z), \neg H(z)\}\}$

Exercício 12.2
Demonstre os teoremas do exercício 8.2, usando resolução SLD e uma função de selecção que escolhe o último literal da cláusula objectivo.
Notas: Como está a tentar provar se uma fórmula é teorema, pode fazer provas por refutação em que as cláusulas correspondem à negação da fórmula inicial. Uma vez que a passagem para a forma clausal já foi feita na aula sobre resolução, apresentam-se com cada teorema as cláusulas que correspondem à sua negação.

1. $(F(a) \wedge \forall x[F(x) \to G(x)]) \to G(a)$
   $\{\{F(a)\}, \{\neg F(x), G(x)\}, \{\neg G(a)\}\}$
2. $(\forall x[F(x) \to G(x)] \wedge \forall y[G(y) \to H(y)]) \to \forall z[F(z) \to H(z)]$
   $\{\{\neg F(x), G(x)\}, \{\neg G(y), H(y)\}, \{F(a)\}, \{\neg H(a)\}\}$
3. $(\forall x[F(x) \to H(x)] \wedge \exists y[F(y)]) \to \exists z[H(z)]$
   $\{\{\neg F(x), H(x)\}, \{F(a)\}, \{\neg H(z)\}\}$

Exercício 12.3
Considere o seguinte conjunto de cláusulas:

- $\{\neg Animal(x), \neg TemPelos(x), Mamifero(x)\}$
- $\{\neg Urso(x), Animal(x)\}$
- $\{\neg Urso(x), TemPelos(x)\}$
- $\{\neg Coelho(x), Mamifero(x)\}$
- $\{Urso(Winnie)\}$
- $\{Coelho(Bugsbunny)\}$
- $\{Animal(Sylvester)\}$
- $\{TemPelos(Sylvester)\}$

Usando resolução SLD e uma função de selecção à sua escolha, responda às seguintes perguntas:

1. O Winnie é mamífero?
2. Quem é que tem pêlos?
3. Quais são os mamíferos?

Exercício 12.4
Considere o seguinte conjunto de cláusulas de Horn:

- $A(x) \leftarrow B(x), C(x)$
- $B(x) \leftarrow D(x)$
- $C(x) \leftarrow E(x)$
- $B(a1) \leftarrow$
- $E(a1) \leftarrow$
- $C(a2) \leftarrow$
- $C(a3) \leftarrow$
- $D(a3) \leftarrow$

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: $\leftarrow A(x)$. Notas: pode usar a estratégia de procura que preferir. No final, indique explicitamente as soluções.

13 Prolog — Árvores de Refutação; Listas

Exercício 13.1
Considere o seguinte programa em Prolog:

```
mamifero(X) :- animal(X), tempelos(X).
animal(X) :- urso(X).
tempelos(X) :- urso(X).
mamifero(X) :- coelho(X).
urso(winnie).
coelho(bugsbunny).
animal(sylvester).
tempelos(sylvester).
```

Mostrando as árvores de refutação respectivas, explique o que é que o Prolog responderia às seguintes perguntas. Considere que são pedidas todas as soluções em cada um dos casos.

1. O Winnie é mamífero?
2. Quem é que tem pêlos?
3. Quais são os mamíferos?

Exercício 13.2
Considere definido da seguinte forma o predicado `m(E, L)`:

```
m(X, [X|_]).
m(X, [_|Xs]) :- m(X, Xs).
```

Mostre as árvores de refutação para os seguintes objectivos, considerando que são pedidas todas as soluções.

1. `m(2, [1, 2, 3]).`
2. `m(X, [1, 2, 3]).`

Exercício 13.3
(Adaptado de "The Art of Prolog", de Leon Sterling e Ehud Shapiro.)
Defina os seguintes predicados que manipulam listas. Em caso de necessidade, em cada alínea pode usar os predicados definidos nas alíneas anteriores.

1. O predicado `membro(Elemento, Lista)`, que tem o valor verdadeiro se `Elemento` for um membro da lista `Lista`. Por exemplo, `membro(2, [1, 2, 3])` tem o valor verdadeiro.

2. O predicado `prefixo(Prefixo, Lista)`, que tem o valor verdadeiro se `Prefixo` for um prefixo da lista `Lista`. Por exemplo, `prefixo([1, 2], [1, 2, 3])` tem o valor verdadeiro.

3. O predicado `sufixo(Sufixo, Lista)`, que tem o valor verdadeiro se `Sufixo` for um sufixo da lista `Lista`. Por exemplo, `sufixo([2, 3], [1, 2, 3])` tem o valor verdadeiro.

4. O predicado `sublista(Sub, Lista)`, que tem o valor verdadeiro se `Sub` for uma sublista da lista `Lista`. Por exemplo, `sublista([2, 3], [1, 2, 3, 4])` tem o valor verdadeiro.

5. O predicado `junta(Xs, Ys, Zs)`, em que a lista `Zs` é o resultado de concatenar as listas `Xs` e `Ys`. Por exemplo, `junta([1, 2], [3, 4], [1, 2, 3, 4])` tem o valor verdadeiro.

6. Redefina os predicados `membro`, `prefixo`, `sufixo` e `sublista` em termos do predicado `junta`.

7. O predicado `seguidos(X, Y, Zs)`, que tem o valor verdadeiro se `X` e `Y` aparecerem seguidos na lista `Zs`. Por exemplo, `seguidos(2, 3, [1, 2, 3])` tem o valor verdadeiro.

8. O predicado `ultimo(X, Xs)`, que tem o valor verdadeiro se `X` for o último elemento da lista `Xs`. Por exemplo, `ultimo(3, [1, 2, 3])` tem o valor verdadeiro.

9. O predicado `inverte(Xs, Ys)`, que tem o valor verdadeiro se `Ys` for uma lista que contém os elementos de `Xs` pela ordem inversa da qual eles aparecem na lista original. Por exemplo, `inverte([1, 2, 3], [3, 2, 1])` tem o valor verdadeiro.

10. O predicado `comprimento(Xs, N)`, que tem o valor verdadeiro se `N` for o comprimento da lista `Xs`. Por exemplo, `comprimento([1], s(0))` tem o valor verdadeiro.

11. O predicado `repete(Xs, XsXs)`, que tem o valor verdadeiro se cada elemento de `Xs` aparece repetido em `XsXs`. Por exemplo, `repete([1, 2], [1, 1, 2, 2])` tem o valor verdadeiro.

14 Prolog — Operadores Pré-definidos

Exercício 14.1
Considere o seguinte programa para calcular o factorial de um número.

```
/*
  factorial1(N, F) :- F e o factorial de N.
*/
factorial1(0, 1).
factorial1(N, F) :- P is N-1, factorial1(P, FP), F is N*FP.
```

1. Qual a sua resposta ao objectivo `factorial1(3, X)`?
2. O que acontece se pedirmos uma segunda solução? Como resolver o problema?
3. Qual a sua resposta ao objectivo `factorial1(X, 6)`? Como resolver o problema?

Exercício 14.2
Escreva o predicado `comp(L, C)`, que tem o valor verdadeiro se `C` é o comprimento da lista `L`.

1. Gerando um processo recursivo.
2. Gerando um processo iterativo.

Exercício 14.3
Escreva o predicado `somalista(Xs, S)`, que tem o valor verdadeiro se `S` corresponde à soma de todos os elementos da lista de inteiros `Xs`.

1. Gerando um processo recursivo.
2. Gerando um processo iterativo.

Exercício 14.4
Escreva o predicado `remove(Xs, X, Ys)`, que tem o valor verdadeiro se `Ys` resulta de remover todas as ocorrências de `X` da lista `Xs`.

Exercício 14.5
Escreva o predicado `escreveLista(Xs)`, que escreve todos os elementos da lista `Xs`, um por linha, e a mensagem `Fim da lista.` no fim da lista.

Exercício 14.6
Suponha que tem uma base de dados que indica as notas que os alunos tiveram nas várias disciplinas (`nota(Nome, Disciplina, Nota)`) e quais os alunos inscritos nas várias disciplinas (`inscrito(Nome, Disciplina)`).

Escreva um programa que permite lançar notas de alunos às disciplinas a que eles estão inscritos e que determina a seguinte informação:

- Lista dos alunos com pelo menos uma nota superior ou igual a um dado valor.
- Média das notas de uma disciplina.
- Média das notas de um aluno.

15 Prolog — Corte; Negação

Exercício 15.1
Escreva um programa que determina o mínimo entre dois números.

1. Usando Prolog puro.
2. Usando cortes.

Exercício 15.2
Explique porque é que o seguinte programa para o `minimo3` não tem os resultados esperados.

```
/*
  minimo3(X, Y, Min) :- Min e o minimo dos numeros X e Y.
*/
minimo3(X, Y, X) :- X =< Y, !.
minimo3(X, Y, Y).
```

Exercício 15.3
Escreva um programa em Prolog que funde duas listas ordenadas de inteiros, tendo como resultado outra lista ordenada de inteiros, incluindo repetições. Por exemplo, `funde([1, 3, 5], [3, 7], [1, 3, 3, 5, 7])` tem o valor verdadeiro.

Exercício 15.4
Em matemática, um polinómio é uma expressão construída a partir de uma ou mais variáveis e constantes, usando apenas os operadores de adição, subtracção e multiplicação, e expoentes inteiros positivos. Por exemplo, `x**2-4*x+7` é a representação de um polinómio em Prolog.

Considere o seguinte programa em Prolog para reconhecer polinómios.

```
/*
  polinomio(Termo, X) :- Termo é um polinomio em X.
*/
polinomio(X, X).
polinomio(Termo, _) :-
   number(Termo).
polinomio(Termo1+Termo2, X) :-
   polinomio(Termo1, X), polinomio(Termo2, X).
polinomio(Termo1-Termo2, X) :-
   polinomio(Termo1, X), polinomio(Termo2, X).
polinomio(Termo1*Termo2, X) :-
   polinomio(Termo1, X), polinomio(Termo2, X).
polinomio(Termo1/Termo2, X) :-
   polinomio(Termo1, X), number(Termo2).
polinomio(Termo**N, X) :-
   integer(N), N >= 0, polinomio(Termo, X).
```

Altere-o, usando cortes verdes, de modo a eliminar ramos desnecessários da árvore de refutação.

Exercício 15.5
Explique qual é o problema com o corte no seguinte programa:

```
/*
  membro3(X, L) :- X e um membro de L.
*/
membro3(X, [X|_]) :- !.
membro3(X, [_|Ys]) :- membro3(X, Ys).
```

Exercício 15.6
Escreva um procedimento `separa(Numeros, Positivos, Negativos)`, em que `Positivos` contém os números positivos da lista de números e `Negativos` contém os números negativos da lista de números. Considere que o zero é um número positivo. Por exemplo, `separa([1, -2, 0, -3], [1, 0], [-2, -3])` tem o valor verdadeiro.

Exercício 15.7
Escreva um predicado `ifThenElse(A, B, C)`, que caso `A` tenha sucesso avalia `B` e caso `A` falhe avalia `C`.

Exercício 15.8
Considere o programa:

```
m(1).
m(2) :- !.
m(3).
```

Diga quais são todas as respostas do Prolog aos seguintes objectivos, considerando que o utilizador escreve **;** até esgotar todas as respostas:

1. `?- m(X).`
2. `?- m(X), m(Y).`
3. `?- m(X), !, m(Y).`
4. `?- m(X), m(Y), !.`
5. `?- m(1), m(2), m(3).`

Exercício 15.9
Considere o seguinte programa em Prolog.

```
d :- a.
d :- g.
a :- b(X), !, c(X).
a :- e.
b(1).
b(2).
c(2).
g.
e.
```

Construa a árvore de refutação para mostrar que `d` sucede através de `g`.

Exercício 15.10
Considere o seguinte programa:

```
p(X, Y) :- q(X, Y).
p(a, b).
q(c, d).
q(e, f).
q(X, Y) :- r(X), !, s(Y).
q(X, Y) :- t(X, Y).
r(e).
r(f).
s(g).
s(h).
t(i, j).
```

Diga quais são as respostas dadas pelo Prolog ao objectivo `p(X, Y)`, considerando que o utilizador escreve `;` até esgotar todas as respostas.

Exercício 15.11
Considere o seguinte programa:

```
u(a).
u(b).

v(1).
v(2).
v(3).

w1(X, Y) :- !, u(X), v(Y).
w2(X, Y) :- u(X), !, v(Y).
w3(X, Y) :- u(X), v(Y), !.
```

Diga quais são as respostas dadas pelo Prolog aos objectivos `w1(X, Y)`, `w2(X, Y)`, e `w3(X, Y)`, considerando que o utilizador escreve `;` até esgotar todas as respostas.

Exercício 15.12
Escreva um predicado de dois argumentos `diferentes`, que tem sucesso apenas quando os seus dois argumentos não são o mesmo, isto é, não são unificáveis.

Exercício 15.13
Defina o conceito de duas listas serem disjuntas usando o `not`, partindo do princípio que existe o predicado `membro/2` que indica se um elemento é membro de uma lista .

Exercício 15.14
Considere o seguinte programa em Prolog.

```
pessoaAlta(X) :- not(baixa(X)), pessoa(X).
pessoa(eva).
pessoa(maria).
baixa(maria).
```

Qual a resposta do Prolog ao objectivo `pessoaAlta(X)`? Corresponde ao que estaria intuitivamente correcto? Se não, explique como é que poderia passar a corresponder.

Exercício 15.15
Considere o seguinte programa:

```
p1(s(X)) :- p1(X).
p2(a).
```

Diga qual a resposta do Prolog ao objectivo `not((p1(X), p2(X)))`.

Exercício 15.16
Considere a seguinte base de conhecimento:

```
cao(bobi).
cao(fiel).
cao(guerreiro).
morde(guerreiro).
```

E as duas formas de representar que o Carlos gosta de todos os cães que não mordam. Repare que em termos lógicos não existem diferenças entre as duas.

```
gosta1(carlos, X) :- cao(X), not(morde(X)).

gosta2(carlos, X) :- not(morde(X)), cao(X).
```

Diga qual a resposta do Prolog a cada um dos objectivos `gosta1(carlos, X)` e `gosta2(carlos, X)`. Se as repostas forem diferentes, explique a razão dessas diferenças.