Introdução a Programação Lógica

Disciplina: Lógica Matemática

Prof. Fermín Alfredo Tang Montané

Curso: Ciência de Computação
Universidade Estadual do Norte Fluminense

Programação Lógica

Predicados

- Definimos sentenças abertas como expressões que dependem de uma ou mais variáveis e que se tornam proposições com valores verdadeiro ou falso quando essas variáveis são substituídas.
- As sentenças abertas são chamadas de **predicados**.
- Apesar de utilizar letras como p e q para denotar predicados, também utiliza-se palavras que ilustram mais claramente o significado do predicado. Por exemplo:

$$casados(x, y)$$

- Para denotar que x é casado com y. Ao invés de uma notação menos ilustrativa como:

$$p_6(x, y)$$

Programação Lógica
Predicados - Exemplos

- **Exemplos:**
- Uma fórmula atómica é um literal positivo.

1.- $p(x)$
2.- $q_2(x, y)$
3.- $mãe(y, x)$
4.- $q_3(x, y, z)$

- A negação de uma fórmula atômica é um literal negativo.

1.- $\neg q_4(x, y)$
2.- $\neg tio(y, x)$
3.- $\neg q_5(x, y, z)$
4.- $\neg avó(x, y)$

Programação Lógica

Implicação

- Em programação lógica, uma implicação do tipo:

$$A \rightarrow B$$

- Costuma ser escrita de maneira inversa:

$$B \leftarrow A$$

- Onde:
- B é chamado de cabeça da implicação, e;
- A é chamado de corpo da implicação.

- A justificativa neste caso, é destacar a conclusão da implicação.

Programação Lógica
Implicação - Exemplos

- **Exemplos:**

1.- $\forall x \, \forall y \;\; (\, p(x,y) \rightarrow q(x,y) \,)$

escrita de maneira inversa: $\forall x \, \forall y \;\; (\, q(x,y) \leftarrow p(x,y) \,)$

onde:

$p(x,y)$ é o corpo;

$q(x,y)$ é a cabeça;

Programação Lógica

Cláusula

- Uma **cláusula** é uma fórmula do cálculo de predicados do seguinte tipo:

$$\forall x_1 , \cdots , \forall x_k \, (L_1 \vee \cdots \vee L_m)$$

- na qual, L_i, onde $1 \leq i \leq m$, é um literal positivo ou negativo. Enquanto, $x_1, \cdots , x_k$, são as variáveis que ocorrem em $L_1 \vee \cdots \vee L_m$.

Programação Lógica

Cláusulas - Exemplos

- **Exemplos:**

1.- $\forall x\ \forall y\ \forall z\ (p(x) \vee q(y,z) \vee \neg p_2(x,y,z))$
2.- $\forall x\ \ p_5(x)$
3.- $\forall y\ \forall z\ (\neg p_6(y) \vee q_3(z))$
4.- $\forall x\ \forall y\ \ (\ tio(x) \vee pai(y)\ )$
5.- $\forall x\ \forall z\ \ (\ maior(x,z) \vee menor(x,z) \vee menor(x,z)\ )$
6.- $\forall x\ (\ solteiro(x) \vee casado(x) \vee viúvo(x) \vee divorciado(x)\ )$

Programação Lógica
Cláusula – Notação Clausal

- A cláusula escrita como:

$$\forall x_1 , \cdots , \forall x_k \, (L_1 \vee \cdots \vee L_m)$$

- está no **formato original de disjunção**. Porém, para escrevermos os programas lógicos, é conveniente escrever a cláusula, na **notação clausal**.
- Escrever uma cláusula na notação clausal é reescrevê-la utilizando-se o conectivo de implicação ("→").
- Os passos para escrever uma clausula na notação clausal são os seguintes:

Programação Lógica
Cláusula – Notação Clausal

- Os passos para escrever uma clausula na notação clausal são os seguintes:
- **Passo 1.-** Separar as literais positivas e negativas. Colocar primeiro as literais negativas seguidas das positivas. (Lei comutativa aplicada ao conectivo de disjunção). Temos assim:

$$\forall x_1 , \cdots , \forall x_k \; (\; (\neg B_1 \vee \cdots \vee \neg B_s) \vee (A_1 \vee \cdots \vee A_r) \;)$$

- **Passo 2.-** Re-escrever as literais negativas, como a negação de uma conjunção. (Lei de DeMorgan). Onde :

$$(\neg B_1 \vee \cdots \vee \neg B_s) \equiv \neg (B_1 \wedge \cdots \wedge B_s)$$

- Temos assim:

$$\forall x_1 , \cdots , \forall x_k \; (\neg (B_1 \wedge \cdots \wedge B_s) \vee (A_1 \vee \cdots \vee A_r) \;)$$

Programação Lógica
Cláusula – Notação Clausal

$$\forall x_1 , \cdots , \forall x_k \, (\neg(B_1 \wedge \cdots \wedge B_s) \vee (A_1 \vee \cdots \vee A_r) \,)$$

- **Passo 3.-** Escrever a cláusula no formato de implicação. Sabe-se que:

$$(A \rightarrow B) \equiv (\neg A \vee B)$$

- Temos assim:

$$\forall x_1 , \cdots , \forall x_k \, (\, (B_1 \wedge \cdots \wedge B_s) \rightarrow (A_1 \vee \cdots \vee A_r) \,)$$

- **Passo 4.-** Escrever a implicação de maneira inversa. **Notação Clausal.**

$$\forall x_1 , \cdots , \forall x_k \, (\, (A_1 \vee \cdots \vee A_r) \leftarrow (B_1 \wedge \cdots \wedge B_s) \,)$$

Programação Lógica

Cláusula – Notação Clausal

$$\forall x_1 , \cdots , \forall x_k \, (\, (A_1 \vee \cdots \vee A_r) \leftarrow (B_1 \wedge \cdots \wedge B_s) \,)$$

- Como todas as variáveis que ocorrem nos literais de uma cláusula estão quantificados universalmente, então não se costuma escrever os quantificadores: $\forall x_1 , \cdots , \forall x_k$.
- Além disso, a cabeça da implicação é sempre uma **disjunção** de literais, assim, em vez de escrever o conectivo " $\vee$ " , escreve-se apenas uma vírgula ",".
- De maneira semelhante, o corpo da implicação é sempre uma **conjunção** de literais, assim, em vez de escrever o conectivo " $\wedge$ " , escreve-se apenas uma vírgula ",".
- Com isso, a clausula pode ser escrita como:

$$A_1, \cdots , A_r \leftarrow B_1, \cdots , B_s$$

- Que é a **notação clausal**.

Programação Lógica
Cláusula de Programa

- Uma cláusula é uma **cláusula de programa**, se:
  - Possuir apenas um literal na cabeça (No formato de disjunção, possui apenas um literal positivo);
  - O corpo pode ser vazio ou não.

- O formato de uma cláusula de programa pode ser:

$$A \leftarrow B_1, \ldots, B_n$$ Corpo não Vazio — Condicional

- Ou:

$$A \leftarrow$$ Corpo Vazio — Incondicional

Programação Lógica

Cláusula de Programa -Exemplos

- **Exemplos:**

1.- $p(x) \leftarrow q(x),\ r(x)$
2.- $r(y) \leftarrow$
3.- $gripe(x) \leftarrow febre(x), mal\_estar(x)$
4.- $q(x) \leftarrow q_1(x,y), q_2(x,z)$
5.- $q_3(z) \leftarrow$
6.- $festa\_ruim(x) \leftarrow pessoal\_chato(x), comida\_ruim(x)$
7.- $aprovado(x) \leftarrow media\_ma\_seis(x), freq\_ok\ (x)$
8.- $r_1(x,y) \leftarrow r_2(x,y), r_3(y,z), r_4(z,w)$
9.- $q_3(x,y,z,w) \leftarrow$

Programação Lógica

Cláusula de Programa Condicional

- Dizemos que uma cláusula de programa é condicional se:
  - A cabeça dela possuir apenas um literal (No formato de disjunção, possui apenas um literal positivo);
  - O corpo da clausula tiver pelo menos um literal. No formato de disjunção, contém pelo menos um literal negativo;

- O formato de uma cláusula de programa condicional é:

$$A \leftarrow B_1, \ldots, B_n$$

Corpo não Vazio

- A interpretação da clausula condicional é a seguinte:

- Para toda atribuição de valores às variáveis que ocorrem na cláusula, se $B_1, \ldots, B_n$ são todas verdadeiras então A é verdadeira.

Programação Lógica

Cláusula de Programa Condicional

- **Exemplos.-**

1.- $p_2(x) \leftarrow p_3(x), q_2(y)$
podemos ler esta cláusula do seguinte modo: "Para toda a atribuição de valores às variáveis x e y, se $p_3(x)$ e $q_2(y)$ forem ambas verdadeiras então $p_2(x)$ também o será.

2.- $p_3(x) \leftarrow q_2(x, y, z)$
podemos ler esta cláusula do seguinte modo: "Para toda a atribuição de valores às variáveis x, y, e z, se $p_3(x)$ for verdadeira então $p_3(x)$ também o será.

3.- $\boldsymbol{tio}(z, x) \leftarrow \boldsymbol{pai}(y, x), \boldsymbol{irmão}(y, z)$
podemos ler esta cláusula do seguinte modo: "Para toda a atribuição de valores às variáveis x, y, e z, se $\boldsymbol{pai}(y, x)$ e $\boldsymbol{irmão}(y, z)$ forem ambas verdadeiras então $\boldsymbol{tio}(z, x)$ também o será.

Programação Lógica

Cláusula de Programa Incondicional

- Dizemos que uma cláusula de programa é incondicional se:
  - A cabeça dela possuir apenas um literal (No formato de disjunção, possui apenas um literal positivo);
  - O corpo dela for vazio. No formato de disjunção, não contém literais negativas;

- O formato de uma cláusula de programa incondicional é:

$$A \leftarrow$$ Corpo Vazio

- A interpretação da clausula incondicional é a seguinte:

- Para toda atribuição de valores às variáveis que ocorrem na cláusula, A é verdadeira.

Programação Lógica

Cláusula de Programa Incondicional

- **Exemplos.-**

1.- $q_2(x) \leftarrow$
podemos ler esta cláusula do seguinte modo: "Para toda a atribuição de valores à variável x, $q_2(x)$ é verdadeira.
2.- $p(x, y, z) \leftarrow$
podemos ler esta cláusula do seguinte modo: "Para toda a atribuição de valores às variáveis x, y, z, $p(x, y, z)$ é verdadeira.
3.- $\boldsymbol{respira}(x) \leftarrow$
podemos ler esta cláusula do seguinte modo: "Para toda a atribuição de valores à variável x, $\boldsymbol{respira}(x)$ é verdadeira. Assumindo que x representa qualquer ser humano, todo ser humano respira.

Programação Lógica

Cláusula Gol

- Dizemos que uma cláusula de programa é uma clausula gol se:
  - A cabeça dela for vazia (No formato de disjunção, não possuir literais positivos);
  - O corpo da clausula tiver pelo menos um literal. No formato de disjunção, contém pelo menos um literal negativo;

- O formato de uma cláusula gol é o seguinte:

$$\leftarrow B_1, \ldots, B_n$$

Cabeça Vazia

Programação Lógica
Cláusula Gol

- O formato de uma cláusula gol é o seguinte:

$$\leftarrow B_1, \ldots, B_n$$

Cabeça Vazia

- No formato de disjunção, sabe-se que uma cláusula gol:

$$\forall x_1, \cdots, \forall x_k \, (\neg B_1 \vee \cdots \vee \neg B_n)$$

- Pela Lei de De Morgan, temos:

$$\forall x_1, \cdots, \forall x_k \, \neg (B_1 \wedge \cdots \wedge B_n)$$

- Pela Equivalência entre quantificadores, sabe-se:

$$(\forall x \, \neg B) \equiv (\neg \exists x \, B)$$

- Com isso, temos:

$$\neg (\exists x_1, \cdots, \exists x_k \, (B_1 \wedge \cdots \wedge B_n))$$

Programação Lógica
Cláusula Gol

- **Exemplos.-**

1.- $\leftarrow\ p(x)$
É o mesmo que escrever: $\neg(\ \exists x\ p(x)\ )$.
2.- $\leftarrow\ p(x), q(y)$
É o mesmo que escrever: $\neg(\ \exists x\ \exists y\ \ p(x) \wedge q(y)\ )$.
3.- $\leftarrow\ q_3(x, y, z)$
É o mesmo que escrever: $\neg(\ \exists x\ \exists y\ \exists z\ \ q_3(x, y, z)\ )$.
4.- $\leftarrow\ \boldsymbol{gripe}(x)$
É o mesmo que escrever: $\neg(\ \exists x\ \boldsymbol{gripe}(x)\ )$.
5.- $\leftarrow\ \boldsymbol{pai}(x, y)$
É o mesmo que escrever: $\neg(\ \exists x\ \exists y\ \boldsymbol{pai}(x, y)\ )$.

Programação Lógica
Cláusula de Horn

- Uma cláusula de Horn é uma clausula de programa (condicional ou incondicional) ou é uma clausula gol.

$$q(x) \leftarrow p(x), p_2(x, y, z), q_4(x, z, y)$$
$$\text{avô}(z, x) \leftarrow pai(z, y), pai(y, x)$$

Programação Lógica
Programas Lógicos

- Introduzimos o conceito de programa lógico e posteriormente os métodos para a computação dos mesmos.
- Um **programa lógico** é um conjunto finito de cláusulas de programa.

Programa P_1

1.- $p(a) \leftarrow$
2.- $q(b) \leftarrow$
3.- $q(x) \leftarrow p(x)$

Programa P_2

1.- $p(a,b) \leftarrow$
2.- $q(a,b) \leftarrow$
3.- $r(x,y) \leftarrow p(x,y)$
4.- $s(x,y) \leftarrow q(y,x)$

Programa P_3

1.- $p(a,b) \leftarrow$
2.- $q(b,c) \leftarrow$
3.- $r(x,z) \leftarrow p(x,y), q(y,z)$

Programa P_4

1.- $p(x) \leftarrow p(x)$
2.- $p(a) \leftarrow$
3.- $q(c) \leftarrow$
4.- $p(b) \leftarrow$

Programação Lógica
Introdução ao PROLOG

- O PROLOG, grosso modo, é o "concretizador" dos programas lógicos:
- Fornecemos a ele um programa lógico e depois podemos fazer várias consultas ao programa.
- O PROLOG ao ler a consulta, a transforma num gol e tenta validar essa clausula no programa por um método de inferência.
- Se conseguir, ele responde "sim!" o que significa que ele conseguiu refutar o
- gol que é o mesmo que dizer que a consulta pode ser deduzida a partir do
- programa.
- Caso contrario ele pode responder "não!" ou inclusive pode não responder nada quando os dados do programa são insuficientes.

Programação Lógica
Notação PROLOG

- A única diferença entre um programa lógico e um programa PROLOG é a notação. Programa lógico P_I

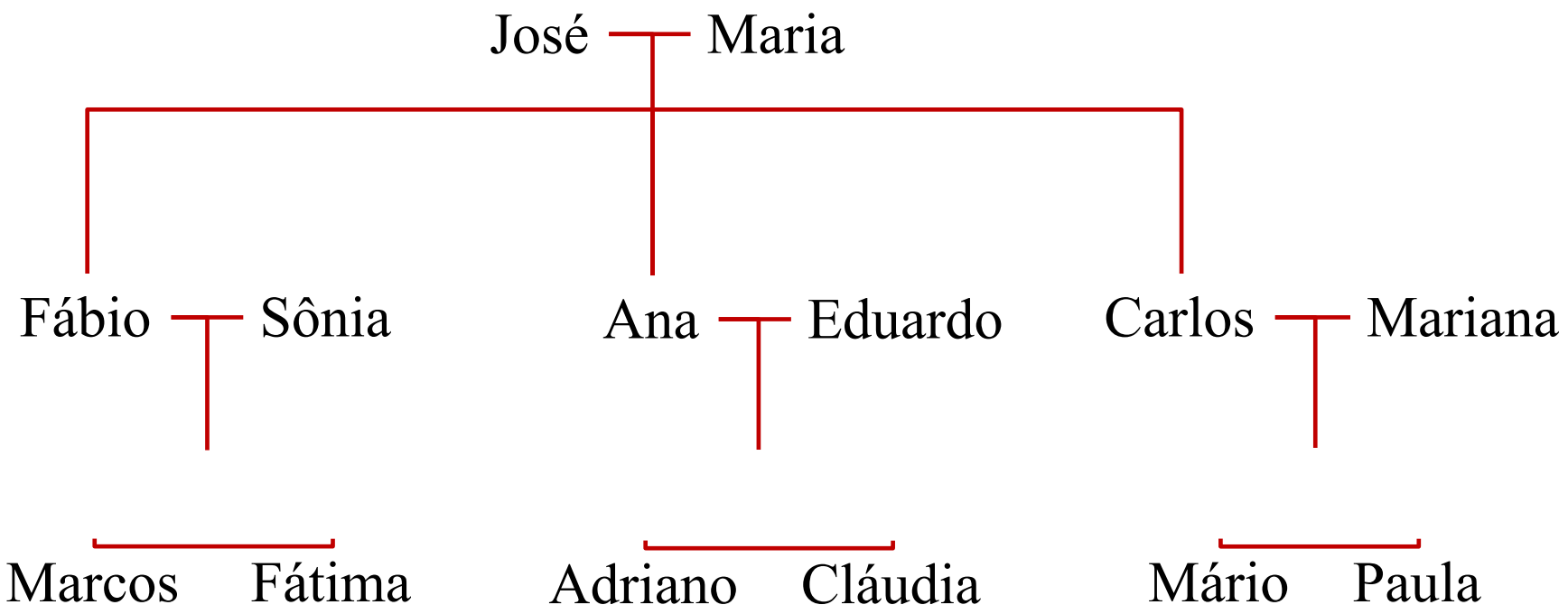
Programa lógico P_I	**Programa P_I (notação de PROLOG)**
$1.\text{-}\ p(a) \leftarrow$	$1.\text{-}\ p(a).$
$2.\text{-}\ q(b) \leftarrow$	$2.\text{-}\ q(b).$
$3.\text{-}\ q(x) \leftarrow p(x)$	$3.\text{-}\ q(X) :- p(X).$

  - Constantes são representadas por letras minúsculas e as variáveis por letras maiúsculas;
  - Todas as cláusulas de programa são finalizadas com ".";
  - Cláusulas de programa incondicionais não necessitam de "←" no final.
  - O conectivo de implicação "←" das cláusulas de programa condicionais é representado por ":-" (dois pontos seguidos de um hífen).
  - As consultas são representadas por: $?-p(a)$
  - Ao invés de: $\leftarrow p(a)$

Programação Lógica

Estudo de Caso 1 – Relações de Parentesco

- Considere a seguinte árvore de família:



Programação Lógica

Estudo de Caso 1 – Relações de Parentesco

- Através de programas lógicos podemos armazenar informações sobre as relações de parentesco das pessoas dentro da família.
- Podemos definir os seguintes predicados:

$$\textbf{\textit{homen}}(x) \equiv \text{“}x \text{ é do sexo masculino”}$$

$$\textbf{\textit{mulher}}(y) \equiv \text{“}y \text{ é do sexo feminino”}$$

- Logo, podemos escrever as seguintes cláusulas de programa (incondicionais) :

1.- $\textbf{\textit{homen}}(José)$.
2.- $\textbf{\textit{homen}}(Fábio)$.
3.- $\textbf{\textit{homen}}(Eduardo)$.
4.- $\textbf{\textit{homen}}(Carlos)$.
5.- $\textbf{\textit{homen}}(Marcos)$.
6.- $\textbf{\textit{homen}}(Adriano)$.
7.- $\textbf{\textit{homen}}(Mário)$.

8.- $\textbf{mulher}(Maria)$.
9.- $\textbf{mulher}(Sônia)$.
10.- $\textbf{mulher}(Ana)$.
11.- $\textbf{mulher}(Mariana)$.
12.- $\textbf{mulher}(Fátima)$.
13.- $\textbf{mulher}(Claudia)$.
14.- $\textbf{mulher}(Paula)$.

Programação Lógica

Estudo de Caso 1 – Relações de Parentesco

- Também, podemos definir as relações, pai e mãe, com os seguintes predicados:

$$\boldsymbol{pai}(x, y) \equiv \text{“}x \text{ é pai de } y\text{”}$$

$$\boldsymbol{mãe}(z, w) \equiv \text{“}z \text{ é mãe de } w\text{”}$$

- Logo, podemos escrever as seguintes cláusulas de programa (incondicionais) :

15.- $\boldsymbol{pai}(José, Fábio)$.
16.- $\boldsymbol{pai}(José, Ana)$.
17.- $\boldsymbol{pai}\ (José, Carlos)$.

18.- $\boldsymbol{pai}(Fábio, Marcos)$.
19.- $\boldsymbol{pai}\ (Fábio, Fátima)$.

20.- $\boldsymbol{pai}\ (Eduardo, Adriano)$.
21.- $\boldsymbol{pai}\ (Eduardo, Cláudia)$.

22.- $\boldsymbol{pai}\ (Carlos, Mário)$.
23.- $\boldsymbol{pai}\ (Carlos, Paula)$.

24.- $\boldsymbol{mãe}(Maria, Fábio)$.
25.- $\boldsymbol{mãe}(Maria, Ana)$.
26.- $\boldsymbol{mãe}(Maria, Carlos)$.

27.- $\boldsymbol{mãe}(Sônia, Marcos)$.
28.- $\boldsymbol{mãe}(Sônia, Fátima)$.

29.- $\boldsymbol{mãe}(Ana, Adriano)$.
30.- $\boldsymbol{mãe}(Ana, Cláudia)$.

31.- $\boldsymbol{mãe}(Mariana, Mário)$.
32.- $\boldsymbol{mãe}(Mariana, Paula)$.

Programação Lógica

Estudo de Caso 1 – Relações de Parentesco

- Tendo essas relações de parentesco básicas estabelecidas podemos construir outros predicados baseados nesses mais básicos. Por exemplo, podemos definir o predicado "irmão", da seguinte maneira:

$$33.\text{-}\ \boldsymbol{irmãos}(x, y) :- \boldsymbol{pai}(z, x), \boldsymbol{pai}(z, y)$$

- Para que duas pessoas x e y serem irmãs, podemos considerar que existe uma pessoa z, tal que z é pai de x e z é pai de y.
- Devemos pensar se está definição é suficiente. Neste caso, estamos assumindo que toda pessoa sempre tem um pai conhecido. Mas poderíamos definir um predicado semelhante utilizando o predicado **mãe**().

Programação Lógica

Estudo de Caso 1: Programa P_1

- Considere o programa P, formado pelas cláusulas (1) – (33).

- Agora podemos realizar a seguinte consulta: **Mário e Paula são irmãos?.** Para isso utilizamos o predicado na forma de uma clausula de gol:

$$?-\boldsymbol{irm\tilde{a}os}(Mário, Paula)$$

- Neste caso teremos uma resposta verdadeira.

Relações de Parentesco

Estudo de Caso 1: Programa P_2

- Podemos estender o programa P_1, formado pelas cláusulas $(1) - (33)$, acrescentando outro predicado. O predicado filho:

$$\boldsymbol{filho}(x, y) \equiv \text{“}x \text{ é filho de } y\text{”}$$

- Pode ser definido da seguinte maneira:

$$34.\text{-}\ \boldsymbol{filho}(x, y) :- \boldsymbol{pai}(y, x), \boldsymbol{homen}(x)$$

- Para que x seja filho de y, y deve ser pai de x e além disso, x deve ser do sexo masculino.

- Agora podemos realizar a seguinte consulta: **Adriano é filho de Eduardo?.** Para isso utilizamos o predicado na forma de uma clausula de gol:

$$?- \boldsymbol{filho}(Adriano, Eduardo)$$

- Neste caso teremos uma resposta verdadeira.

Programação Lógica
Estudo de Caso 2

- **Alice, o Leão e o Unicórnio.-**
- No livro "Alice no país das maravilhas" o autor Lewis Carol, que era um lógico, incorpora certo conteúdo de lógica por trás do texto. A passagem do texto é a seguinte:
- Alice entra em uma floresta e perde a noção dos dias da semana. O Leão e o Unicórnio eram duas das criaturas que habitavam a floresta. Alice os encontra e deseja obter alguma informação sobre o dia da semana. O problema é que o Leão mente em determinados dias da semana e o mesmo acontece com o Unicórnio. Alice sabe que eles mentem e sabe em que dia cada um mente. Naquela época o Leão mentia às segundas, terças, sábado e domingo e falava a verdade nos outros dias da semana. O Unicórnio mentia às segundas, quartas, sextas e domingos e falava a verdade nos outros dias da semana.
- Quando Alice os encontra
- o Leão diz: **Ontem foi um dos meus dias de mentir!**
- e o Unicórnio diz: **Ontem foi um dos meus dias de mentir!**
- A partir dessas informações, Alice descobriu o dia da semana. Qual era?

Programação Lógica
Estudo de Caso 2

- **Alice, o Leão e o Unicórnio.-**
- A ideia é implementar um programa lógico que dadas as informações sobre os dias que o Leão e o Unicórnio mentem, o programa determina em que dia Alice, o Leão e o Unicórnio se encontraram.

- Primeiro devemos criar um predicado **ontem** cuja a função é fornecer a sequência dos dias da semana para o programa. Este predicado tem o seguinte formato:

$$\boldsymbol{ontem}(x, y) \equiv \text{“}y \text{ é o dia da semana anterior ao dia } x\text{”}$$

- Com isso, podemos escrever:

$$\boldsymbol{ontem}(domingo, sábado).$$
$$\boldsymbol{ontem}(segunda, domingo).$$
$$\boldsymbol{ontem}(terça, segunda).$$
$$\boldsymbol{ontem}(quarta, terça).$$
$$\boldsymbol{ontem}(quinta, quarta).$$
$$\boldsymbol{ontem}(sexta, quinta).$$
$$\boldsymbol{ontem}(sábado, sexta).$$

Programação Lógica
Estudo de Caso 2

- **Alice, o Leão e o Unicórnio.-**
- Depois, devemos criar um predicado que relacione o Leão e o Unicórnio aos dias em que eles mentem. O predicado **mentira** faz este serviço:

$$\boldsymbol{mentira}(x, y) \equiv \text{“}x \text{ mente no dia } y\text{”}$$

- Onde, a variável x pode ser o Leão e o Unicórnio. A variável y é um dos dias da semana. De acordo com os dados do texto, podemos escrever o seguinte:

$$\boldsymbol{mentira}(leão, segunda).$$
$$\boldsymbol{mentira}(leão, terça).$$
$$\boldsymbol{mentira}(leão, sábado).$$
$$\boldsymbol{mentira}(leão, domingo).$$
$$\boldsymbol{mentira}(unicórnio, segunda).$$
$$\boldsymbol{mentira}(unicórnio, quarta).$$
$$\boldsymbol{mentira}(unicórnio, sexta).$$
$$\boldsymbol{mentira}(unicórnio, domingo).$$

Programação Lógica

Estudo de Caso 2 - Alice, o Leão e o Unicórnio

- O programa completo é o seguinte:

$$\begin{aligned}
&\boldsymbol{ontem}(domingo, sábado).\\
&\boldsymbol{ontem}(segunda, domingo).\\
&\boldsymbol{ontem}(terça, segunda).\\
&\boldsymbol{ontem}(quarta, terça).\\
&\boldsymbol{ontem}(quinta, quarta).\\
&\boldsymbol{ontem}(sexta, quinta).\\
&\boldsymbol{ontem}(sábado, sexta).\\
&\boldsymbol{mentira}(leão, segunda).\\
&\boldsymbol{mentira}(leão, terça).\\
&\boldsymbol{mentira}(leão, sábado).\\
&\boldsymbol{mentira}(leão, domingo).\\
&\boldsymbol{mentira}(unicórnio, segunda).\\
&\boldsymbol{mentira}(unicórnio, quarta).\\
&\boldsymbol{mentira}(unicórnio, sexta).\\
&\boldsymbol{mentira}(unicórnio, domingo).
\end{aligned}$$

- Este programa é suficiente para Alice determinar o dia em que ela conversou com o Leão e o Unicórnio porém devemos fazer as consultas corretamente para obtermos alguma resposta.

Programação Lógica

Estudo de Caso 2 - Alice, o Leão e o Unicórnio

- Quando o Leão (ou o Unicórnio) diz que o dia anterior era um dos dias dele mentir ele pode estar mentindo. Como temos o Leão e o Unicórnio, temos quatro possibilidades:
  - i) os dois podem estar mentindo; ii) o leão mentindo e o unicórnio não;
  - iii) o unicórnio mentindo e o leão não; iv) os dois podem estar falando a verdade.

- Considere que x é o dia em que Alice encontra com o Leão e o Unicórnio e y é o dia anterior. **Queremos saber que dia foi y ?.** Para isso realizamos consultas com predicados da seguinte maneira:

$$?-\boldsymbol{ontem}(x,y),\boldsymbol{mentira}(leão,y),\boldsymbol{verdade}(leão,x)$$

- Quais são os dias x e y, sendo que no dia x, o leão disse a verdade sobre o dia y. Ou seja, no dia x em que encontro com Alice disse a verdade e no dia anterior y mentiu.

- Realizamos consultas com três predicados e aplicamos a regra de resolução:

Programação Lógica

Estudo de Caso - Alice, o Leão e o Unicórnio

- i) O leão está mentindo no dia x, e disse a verdade no dia y, anterior.

  $?-\boldsymbol{ontem}(x,y),\boldsymbol{verdade}(leão,y),\boldsymbol{mentira}(leão,x)$

  Obtemos como resultado $x =$ sábado e y $=$ sexta.

- ii) O unicórnio está mentindo no dia x, e disse a verdade no dia y, anterior.

  $?-\boldsymbol{ontem}(x,y),\boldsymbol{verdade}(unicórnio,y),\boldsymbol{mentira}(unicórnio,x)$

  Obtemos como resultado $x =$ domingo e y $=$ sábado.

  $x =$ sexta e y $=$ quinta.

  $x =$ quarta e y $=$ terça.

- iii) O leão está falando a verdade no dia x, e mentindo no dia y, anterior.

  $?-\boldsymbol{ontem}(x,y),\boldsymbol{mentira}(leão,y),\boldsymbol{verdade}(leão,x)$

  Obtemos como resultado $x =$ quarta e y $=$ terça.

- iv) O unicórnio está falando a verdade no dia x, e mentindo no dia y, anterior.

  $?-\boldsymbol{ontem}(x,y),\boldsymbol{mentira}(unicórnio,y),\boldsymbol{verdade}(unicórnio,x)$

  Obtemos como resultado $x =$ terça e y $=$ segunda.

  $x =$ quinta e y $=$ quarta.

  $x =$ sábado e y $=$ sexta.

Programação Lógica

Estudo de Caso - Alice, o Leão e o Unicórnio

	Seg	Ter	Qua	Qui	Sex	Sab	Dom
Leão	X	X				X	X
Unicórnio	X		X		X		X

- Quando Alice os encontra:
  - o Leão diz: **Ontem foi um dos meus dias de mentir!**
  - e o Unicórnio diz: **Ontem foi um dos meus dias de mentir!**
- Qual é o dia da semana?

- i) os dois podem estar mentindo;

  Na segunda ou não domingo. Más é impossível, porque o dia anterior seria de falar a verdade.
- ii) o leão mentindo e o unicórnio não;

  É sábado!!!
- iii) o unicórnio mentindo e o leão não;

  É quarta!!!
- iv) os dois podem estar falando a verdade.

  Na quarta, mas é impossível.

Referência

- **Abe,** Jair Minoro**; Scalzitti** Alexandre **e Silva Filho,** João Inácio. Introdução a Lógica para Ciência da Computação. Capítulo 3. 1ª Edição. Editora Arte Ciência. São Paulo. 2001.