



Representação de Informação Incompleta

Unidade de ensino, 2002

Cesar Analide, Paulo Novais, José Neves



**Departamento de Informática
Grupo de Inteligência Artificial**

Objectivos

- Representação de conhecimento imperfeito;
- Representação simbólica de informação incompleta, inconsistente, incerta, etc.;
- Implementação de mecanismos de raciocínio não-monótono.

Motivação

- Manipulação de informação simbólica;
- Representação explícita de informação falsa;
- Extensão da capacidade das respostas a questões.

BD's *versus* RC

- Bases de Dados:

- Pressuposto dos Nomes Únicos;
- Pressuposto do Mundo **Fechado**;
- Pressuposto do Domínio **Fechado**.

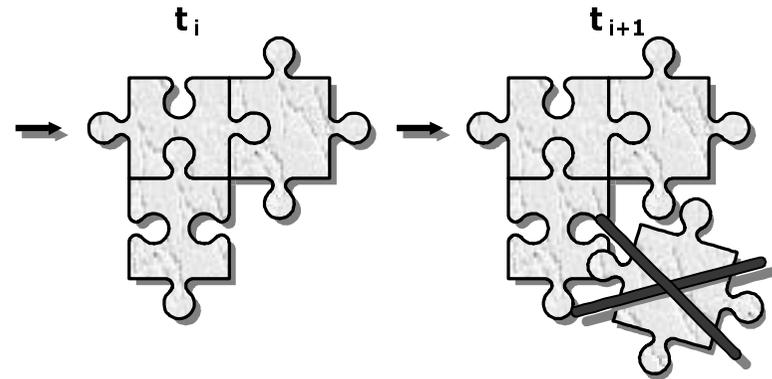
- Representação de Conhecimento:

- Pressuposto dos Nomes Únicos;
- Pressuposto do Mundo **Aberto**;
- Pressuposto do Domínio **Aberto**.

Monotonia vs Não-Monotonia

● Monotonia:

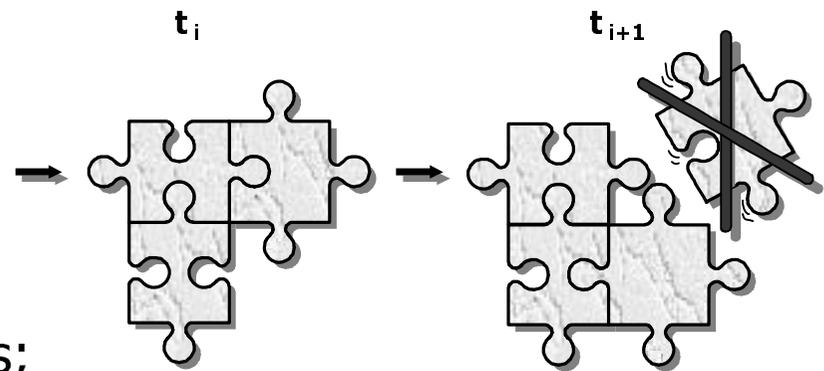
Não é permitida informação que contrarie conclusões anteriores.



● Não-Monotonia:

Justifica-se pela:

- consideração de pressupostos temporários;
- obtenção de conclusões plausíveis;
- flexibilização da evolução do conhecimento;
- dificuldade na representação completa do conhecimento.



Programação em Lógica

- A programação em lógica determina a veracidade ou falsidade de questões:

$\text{voa}(X) \leftarrow \text{ave}(X)$

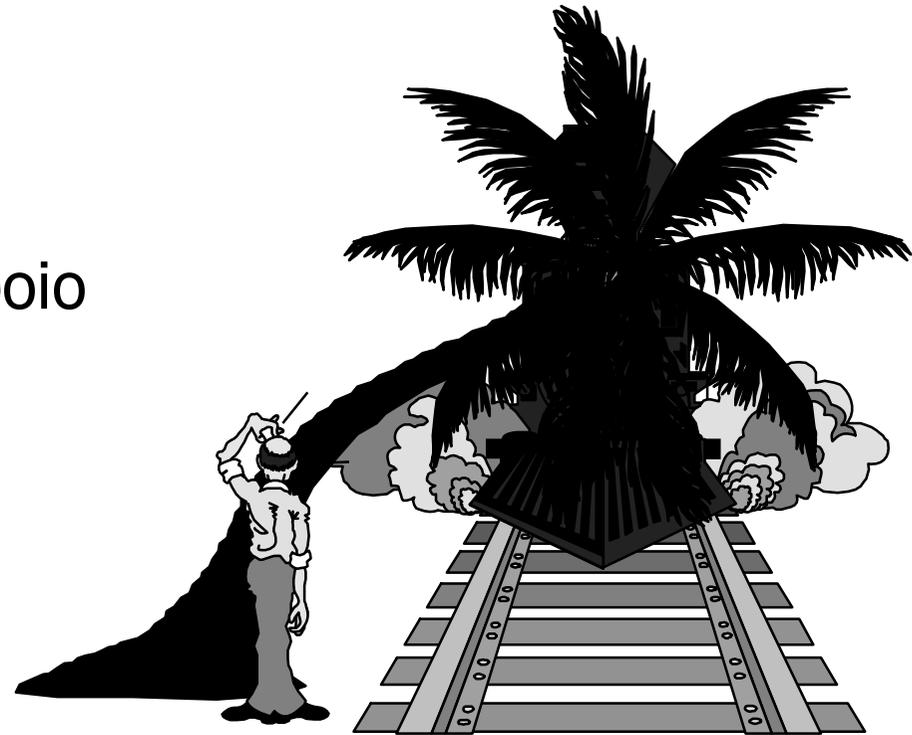
$\text{n\~{a}o-voa}(X) \leftarrow \text{avestruz}(X)$

- A extensão à programação em lógica permite representar explicitamente informação falsa:

$\neg\text{voa}(X) \leftarrow \text{avestruz}(X)$

A Negação na PLE

- Negação por falha na prova:
atravessar \leftarrow não comboio
- Negação forte:
atravessar \leftarrow \neg comboio



Interpretação na PLE

- Genericamente, a resposta a uma questão $q(X)$ é:

- ***verdadeira*** se

$$\exists_x : q(X)$$



- ***falsa*** se

$$\exists_x : \neg q(X)$$



- ***desconhecida*** se

$$\neg \exists_x : q(X) \vee \neg q(X)$$



O PMF na PLE

• Aplicação do PMF na PLE:

par(0)

par(s(s(X))) ← par(X)

O conjunto de soluções é:

{par(0), par(2), ...}

o que implica par(1) ser
desconhecido

par(0)

par(s(s(X))) ← par(X)

¬par(X) ← não par(X)

O conjunto de soluções é:

{par(0), ¬par(1), par(2), ...}

o que implica par(1) ser
falso

Informação Incompleta

- Valores Nulos:
 - Tipo Desconhecido; **I**
 - Tipo Desconhecido, mas de um conjunto determinado de hipóteses; **II**
 - Tipo Desconhecido e não permitido. **III**

Valores Nulos I

Filho F	P
João	Adão
Belém	<u>Alguém</u>

filho(joão, adão)

filho(belém, alguém)

\neg filho(F, P) \leftarrow não filho(F, P) \wedge
não exceção(F, P)

exceção(F, P) \leftarrow filho(F, alguém)

- A Belém é filha de quem?
- A Belém é filha do Adão?

filho(belém, X)

filho(belém, adão)

Valores Nulos II

Filho F	P
João	Adão
Belém	<u>Alguém</u>
Maria	{ <u>Faria</u> , <u>Garcia</u> }

filho(joão, adão)

filho(belém, alguém)

\neg filho(F, P) \leftarrow não filho(F, P) \wedge
não exceção(F, P)

exceção(F, P) \leftarrow filho(F, alguém)

exceção(maria, faria)

exceção(maria, garcia)

- A Maria é filha do Sofia?
- A Maria é filha do Faria?

filho(maria, sofia)

filho(maria, faria)

Valores Nulos III

Filho F	P
João	Adão
Belém	<u>Alguém</u>
Maria	{ <u>Faria</u> , <u>Garcia</u> }
<u>βεβε</u>	Pelé

- O André é filho do Pelé?

...
 $\neg \text{filho}(F, P) \leftarrow \text{não filho}(F, P) \wedge$
 $\text{não exceção}(F, P)$

$\text{exceção}(F, P) \leftarrow \dots$

...
 $\text{filho}(\underline{\beta\epsilon\beta\epsilon}, \text{pelé})$
 $\text{exceção}(F, P) \leftarrow \text{filho}(\underline{\beta\epsilon\beta\epsilon}, P)$

$\text{nulo}(\underline{\beta\epsilon\beta\epsilon})$
 $\leftarrow \text{filho}(F, \text{pelé}) \wedge \text{não nulo}(F)$

$\text{filho}(\text{andré}, \text{pelé})$

Interpretador

demo: questão \times resposta $\rightarrow \{V, F\}$

verdadeira

$$\exists_x : q(X)$$

demo(Q, verdadeiro) : -
Q.

falsa

$$\exists_x : \neg q(X)$$

demo(Q, falso) : -
 $\neg Q$.

desconhecida

$$\neg \exists_x : q(X) \vee \neg q(X)$$

demo(Q, desconhecido) : -
não Q, não $\neg Q$.

Questões: demo(Q,R)

filho(joão, adão)

filho(belém, alguém)

filho(βεβε, pelé)

\neg filho(F,P) \leftarrow não filho(F,P) \wedge
não exceção(F,P)

exceção(F,P) \leftarrow filho(F, alguém)

exceção(maria, faria)

exceção(maria, garcia)

exceção(F, P) \leftarrow filho(βεβε, P)

nulo(βεβε)

\leftarrow filho(F, pelé) \wedge não nulo(F)

• demo(filho(belém, X), R) ✓

• demo(filho(belém, adão), R) ?

• demo(filho(maria, sofia), R) ✗

• demo(filho(maria, faria), R) ?

• demo(filho(andré, pelé), R) ?

Conclusões

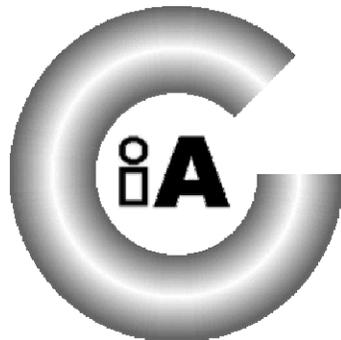
- Extensão à Programação em Lógica:
 - Duas formas de negação;
 - Distinção entre falso e não verdadeiro.
- Formalização do PMF na PLE:
 - Maior flexibilidade;
 - Identificação, tratamento e raciocínio sobre valores nulos.
- Novo tipo de dados: **Valores Nulos.**

Sugestões

- Sofisticação do interpretador;
- Tratamento da assimilação de conhecimento e aprendizagem;
- Manipulação de bases de conhecimento não destrutivas.

Apresentação

- Cesar Analide
analide@di.uminho.pt
- Paulo Novais
pjon@di.uminho.pt
- José Neves
jneves@di.uminho.pt



Grupo de Inteligência Artificial
Departamento de Informática
Universidade do Minho