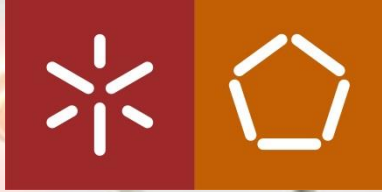




**Universidade do Minho**  
Escola de Engenharia  
Departamento de Informática

# Representação de Conhecimento Imperfeito

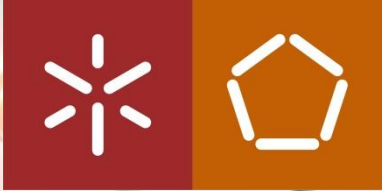
**Cesar Analide, Paulo Novais, José Neves**



# Representação de Conhecimento Imperfeito

## Objetivos

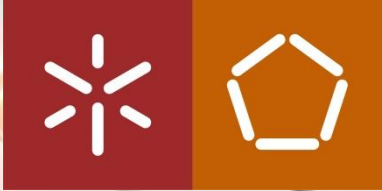
- Representação de conhecimento imperfeito;
- Representação simbólica de informação incerta, imprecisa, interdita, e outras;
- Implementação de mecanismos de raciocínio não-monótono.



# Representação de Conhecimento Imperfeito

## Motivação

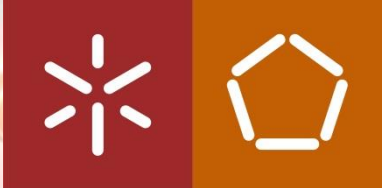
- Manipulação de informação simbólica;
- Representação explícita de conhecimento negativo;
- Extensão da capacidade de responder a perguntas;
- Expansão da habilidade para resolver problemas.



# Representação de Conhecimento Imperfeito

**Bases de dados**  
*versus*  
**Bases de**  
**conhecimento**

- Bases de Dados:
  - Pressuposto dos Nomes Únicos;
  - Pressuposto do **Mundo Fechado**;
  - Pressuposto do **Domínio Fechado**.
  
- Bases de Conhecimento:
  - Pressuposto dos Nomes Únicos;
  - Pressuposto do **Mundo Aberto**;
  - Pressuposto do **Domínio Aberto**.

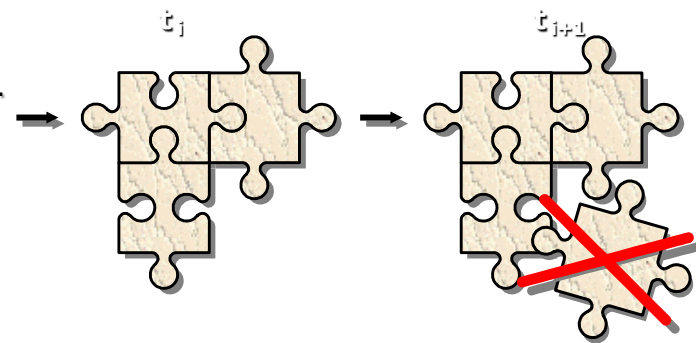


# Representação de Conhecimento Imperfeito

## Monotonia *versus* não monotonia

### ■ Monotonia:

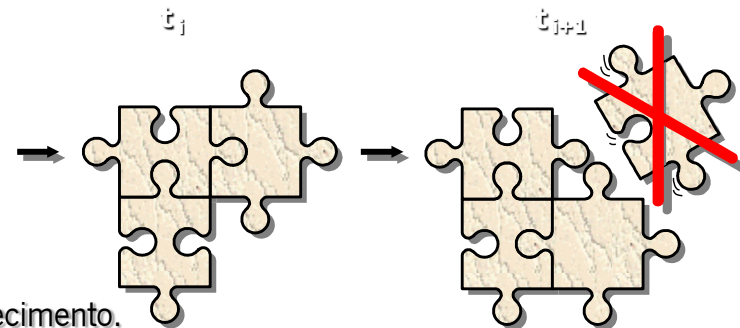
- Não admite contradição com conclusões anteriores.



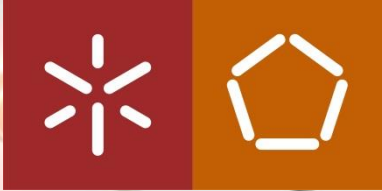
### ■ Não monotonia:

Justifica-se pela

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



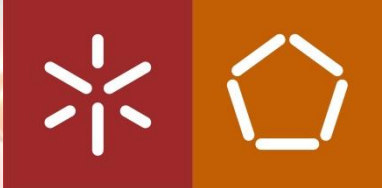




# Representação de Conhecimento Imperfeito

## Programação em lógica

- A programação em lógica determina a veracidade ou falsidade de questões:
  - $\text{voa}(X) \leftarrow \text{ave}(X)$
  - $\text{n\~ao-voa}(X) \leftarrow \text{avestruz}(X)$



# Representação de Conhecimento Imperfeito

## Programação em lógica estendida (PLE)

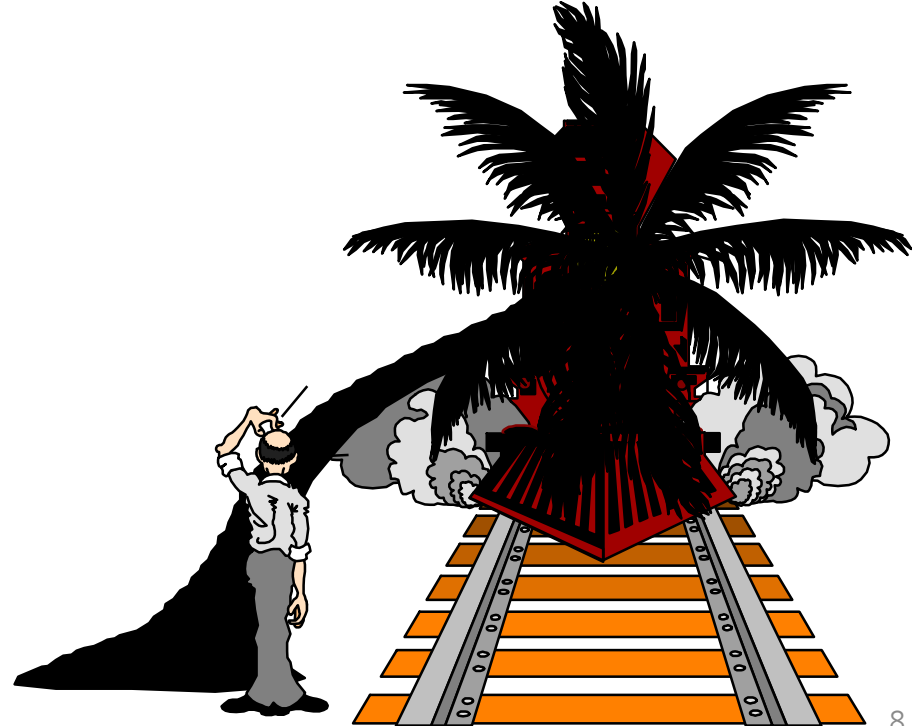
- A programação em lógica determina a veracidade ou falsidade de questões:
  - $\text{voa}(X) \leftarrow \text{ave}(X)$
  - $\text{não-voa}(X) \leftarrow \text{avestruz}(X)$
- A extensão à programação em lógica permite representar explicitamente conhecimento negativo:
  - $\neg \text{voa}(X) \leftarrow \text{avestruz}(X)$



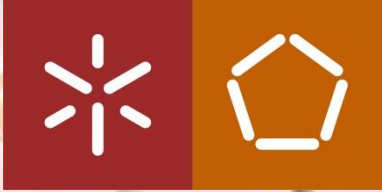
# Representação de Conhecimento Imperfeito

## A negação

- Negação por falha na prova:
  - atravessar ← não comboio



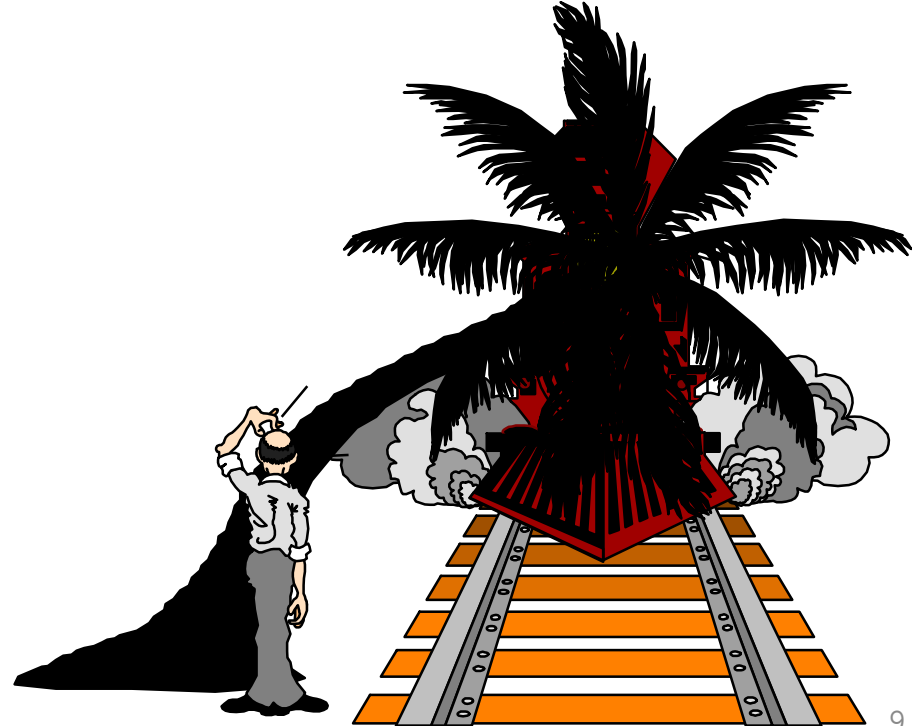


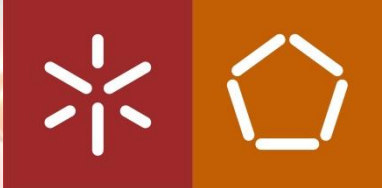


# Representação de Conhecimento Imperfeito

## A negação

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





# Representação de Conhecimento Imperfeito

## Interpretação

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

○ **verdadeira** se

$$\exists X : q(X)$$



○ **falsa** se

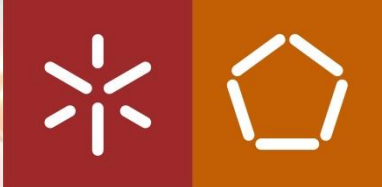
$$\exists X : \neg q(X)$$



○ **desconhecida** se

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





# Representação de Conhecimento Imperfeito

## O Pressuposto do Mundo Fechado

### ▪ Aplicação do PMF na PLE:

- $\text{par}(0)$
- $\text{par}(s(s(X))) \leftarrow \text{par}(X)$

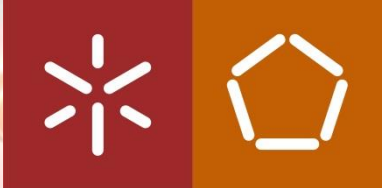
- O conjunto de soluções é:  
 $\{\text{par}(0), \text{par}(2), \text{par}(4), \dots\}$

o que permite  $\text{par}(1)$  ser **desconhecido**

- $\text{par}(0)$
- $\text{par}(s(s(X))) \leftarrow \text{par}(X)$
- $\neg \text{par}(X) \leftarrow \text{não par}(X)$

- O conjunto de soluções é:  
 $\{\text{par}(0), \neg \text{par}(1), \text{par}(2), \neg \text{par}(3), \dots\}$

o que permite  $\text{par}(1)$  ser **falso**



# Representação de Conhecimento Imperfeito

## Informação Incompleta

### ■ Valores Nulos:

- Incerto

Desconhecido;



- Impreciso

Desconhecido, mas de um conjunto determinado de hipóteses;



- Interdito

Desconhecido e não permitido conhecer.





# Representação de Conhecimento Imperfeito

## Base de conhecimento

### ■ Relação Filho: Filhos x Pais

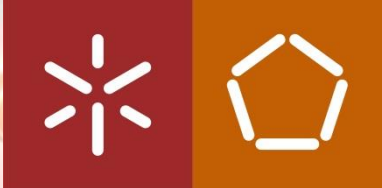
Filhos	Pais
João	José
José	Manuel
Carlos	José

- filho( joao,jose ).
- filho( jose,manuel ).
- filho( carlos,jose ).
- 

- O João é filho do José?
- O João é filho do Adão?

- filho( joao,jose )?
- filho( joao,adao )?





# Representação de Conhecimento Imperfeito

## Formalização do PMF

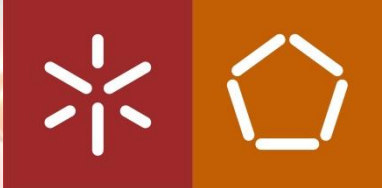
### ▪ Relação Filho: Filhos x Pais

Filhos	Pais
João	José
José	Manuel
Carlos	José

- filho( joao,jose ).
- filho( jose,manuel ).
- filho( carlos,jose ).
- $\neg \text{filho}( F,P ) \leftarrow \text{não filho}( F,P ) \wedge \text{não exceção}( F,P )$

- O João é filho do José?
- O João é filho do Adão?

- filho( joao,jose )?
- filho( joao,adao )?



# Representação de Conhecimento Imperfeito

## Valores nulos

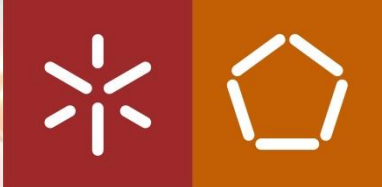


### ■ Relação Filho: Filhos x Pais

Filhos	Pais
João	José
José	Manuel
Carlos	José
Belém	Alguém

- A Belém é filha do Adão?
- A Belém é filha do João?

- filho( joao,jose ).
- filho( jose,manuel ).
- filho( carlos,jose ).
- $\neg \text{filho}( F,P ) \leftarrow \text{não filho}( F,P ) \wedge \text{não exceção}( F,P )$
- filho( belém, alguém)
- exceção( F,P )  $\leftarrow$  filho( F,alguém )
- filho( belem,adao )?
- filho( belem,joao )?



# Representação de Conhecimento Imperfeito

## Valores nulos



### ■ Relação Filho: Filhos x Pais

Filhos	Pais
João	José
José	Manuel
Carlos	José
Belém	Alguém
Maria	{ Faria, Garcia }

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

- filho( joao,jose ).
- filho( jose,manuel ).
- filho( carlos,jose ).
- $\neg \text{filho}( F,P ) \leftarrow \text{não filho}( F,P ) \wedge \text{não exceção}( F,P )$
- ...
- exceção( maria,faria ).
- exceção( maria,garcia ).
- filho( maria,faria )?
- filho( maria,sofia )?



# Representação de Conhecimento Imperfeito

## Valores nulos



### Relação Filho: Filhos x Pais

Filhos	Pais
João	José
José	Manuel
Carlos	José
Belém	Alguém
Maria	{ Faria, Garcia }
Bebé	Júlio

- O André é filho do Júlio?
- O André é filho do Júlio.

- ...
- $\neg \text{filho}(F,P) \leftarrow \text{não filho}(F,P) \wedge \text{não exceção}(F,P).$
- $\text{filho}(\text{bebe}, \text{julio}).$
- $\text{nulo}(\text{bebe}).$
- $\text{exceção}(F,P) \leftarrow \text{filho}(\text{bebe}, P).$
- $\leftarrow \text{filho}(F, \text{julio}) \wedge \text{não nulo}(F)$
- $\text{filho}(\text{andre}, \text{julio}).$
- $\text{filho}(\text{andre}, \text{julio}).$



# Representação de Conhecimento Imperfeito

## Interpretador

- Meta-interpretador:  
demo: Questão x Resposta

- **verdadeira**

$$\exists X : q(X)$$

- **falsa**

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

- **desconhecida**

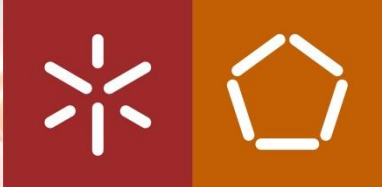
$$\neg \exists X : q(X) \vee \neg q(X)$$

- `demo( Questao, verdadeiro ) :-  
Questao.`

- `demo( Questao, falso ) :-  
-Questao.`

- `demo( Questao, desconhecido ) :-  
nao( Questao ),  
nao( -Questao ).`

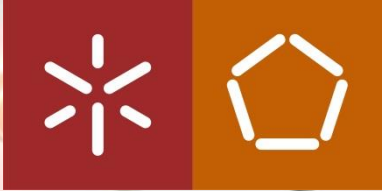




# Representação de Conhecimento Imperfeito

## Conclusões

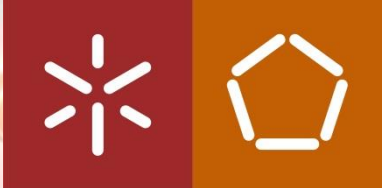
- Extensão à Programação em Lógica:
  - Duas formas de negação;
  - Distinção entre falso e não verdadeiro.
- Formalização do PMF na PLE:
  - Maior flexibilidade;
  - Identificação, tratamento e raciocínio sobre valores nulos.
- Novo tipo de dados: Valores Nulos.



# Representação de Conhecimento Imperfeito

## **Outros desenvolvimentos**

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



# Representação de Conhecimento Imperfeito

## Sofisticação do interpretador

- Demonstração de questões dadas por:
  - Sequência de questões:
    - “Quem são os filhos do José, do Manuel e da Maria?”
  - Agregação de questões:
    - “O filho do José é filho da Maria?”
    - “O filho do José é neto do António?”
- Agregações de questões podem ser:
  - Dadas pela conjunção de questões;
  - Dadas pela disjunção de questões;
  - Dadas pela conjunção e disjunção de questões.



# Representação de Conhecimento Imperfeito

## **Referências bibliográficas**

- Cesar Analide, José Neves, “Representação de Informação Incompleta”, Texto Pedagógico, 2010.
- Ivan Bratko, "PROLOG: Programming for Artificial Intelligence", 3rd Edition, Addison-Wesley Longman Publishing Co., Inc., 2000.
- Hélder Coelho, "A Inteligência Artificial em 25 lições", Fundação Calouste Gulbenkian, 1995.



**Intelligent Systems Lab**

**Contactos**

- Universidade do Minho
- Escola de Engenharia
- Departamento de Informática
- <http://islab.di.uminho.pt>
- DI-3.22