

# Lógica Computacional

---

Apresentação da Disciplina

Introdução

Inferência

Linguagem Natural e Linguagens de 1ª Ordem

Domínios de Aplicação

# Introdução à Cadeira

---

## Objectivos da Disciplina

A disciplina de Lógica Computacional, é uma unidade curricular obrigatória da LEI (2º ano, 1º semestre), com 6 unidades de crédito ECTS.

Na disciplina é feita uma introdução à **Lógica Proposicional e à Lógica de Predicados de primeira ordem**, focada nas noções de linguagem formal e de argumentação bem como da sua formalização em sistemas de dedução, sendo estudados 2 desses sistemas (dedução natural e resolução).

## Livro de Texto principal

Language Proof and Logic (2nd edition).

Dave Barker-Plummer, Jon Barwise and John Etchemendy.

CSLI Publications, 2011.

## Página da cadeira

[ssdi.di.fct.unl.pt/lc](http://ssdi.di.fct.unl.pt/lc)

# Lógica Computacional - Introdução

---

- Inteligência e Raciocínio
- Raciocínio e Pensamento Simbólico
- Símbolos e Linguagem
- Linguagens Simbólicas
  - Linguagem Natural – Português, Francês, Inglês, ...
  - Linguagem da Matemática (números, operações, funções, relações, ...)
  - Linguagens de Programação (Máquina, Assembler, Java, C, Prolog, ...)

# Automatização do Raciocínio

---

## - Inferência

- A partir de premissas (factos ou hipóteses) tirar conclusões

## - Qual a validade da conclusão?

- O que significa validade?

## - Tipos de Inferência

- Dedução
- Indução
- Abdução
- Indução Estrutural

# Automatização do Raciocínio

---

## Dedução

As bolas da caixa são todas pretas  
Esta bola saiu da caixa  
∴ Esta bola é preta

## Abdução

As bolas da caixa são todas pretas  
Esta bola é preta  
∴ Esta bola saiu da caixa

## Indução

Esta bola saiu da caixa e é preta  
Esta outra também  
E esta ainda  
...  
∴ Todas as bolas da caixa são pretas  
∴ Todas as bolas pretas são da caixa

## Indução Estrutural

Esta bola saiu da caixa e é preta  
A cor de cada bola é igual à anterior  
∴ Todas as bolas da caixa são pretas

# Linguagem Natural

---

- Ambiguidade e Expressividade
  - A Maria é alta - A casa é alta
  - A Maria saltou à corda. Ela era muito velha.
- Quem desambigua?
  - Seres inteligentes com muito conhecimento de contexto
- Automatização do Raciocínio
  - Linguagens menos ambíguas
  - Menos Expressividade
  - Maior facilidade em encadear os raciocínios

# Linguagens de 1ª Ordem

---

- Linguagens de 1ª Ordem

- Constantes
- Predicados
- Funções

- Exemplos

- Linguagem Natural (restringida)
- Aritmética
- Teoria de Conjuntos
- O Mundo de Tarski

# Linguagens de 1ª Ordem

---

Começamos com frases/proposições **simples**. Estas frases traduzem-se numa **fórmula de 1ª ordem com um só predicado**.

- A Maria é alta. O João conhece o pai do Pedro.
  - Constantes: {joão, maria, pedro}
  - Predicados: {Alta/1, Conhece/2}
  - Funções: {pai/1}

A Maria é alta
Alta(maria)

O João conhece o pai do Pedro
Conhece(joão,pai(pedro))

Mas

- A Maria conhece um amigo do João
  - Requer uma representação mais complexa – “amigos há muitos”

**Nota:** Para cada domínio de aplicação, deverá ser escolhida uma **assinatura**, que é o conjunto das constantes, funções e predicados a utilizar.

# Regras de Tradução

---

- Constantes denotam “objectos”.
- Uma constante denota um só objecto. Um objecto pode ser denotado por 0 ou mais constantes.
- Funções denotam objectos. Os seus argumentos são constantes ou outras funções.
- **Nota:** Uma constante é uma função sem argumentos.
- Uma função com os seus argumentos não pode denotar dois objectos.
  - pai(joão) ok mas amigo(joão) ko
- A ordem dos argumentos (em predicados e funções) **não** é arbitrária.
  - A caneta que o Rui deu ao Nuno vs. a caneta que o Nuno deu ao Rui  
o caneta(rui, nuno) vs. caneta(nuno, rui)
- **Convenção:** Constantes e funções começam com minúsculas.

# Regras de Tradução

---

- Predicados correspondem
  - a propriedades de objectos
    - o A Maria é alta Alta(maria)
  - ou relações entre objectos
    - o A Maria conhece o Pedro. Conhece(maria, pedro)
    - o A Maria viu o Pedro na aula Ver(maria, pedro, aula)
- **Convenção:** Predicados começam com maiúsculas.
- A ordem dos argumentos em predicados **não** é arbitrária.
  - O Rui é mais alto que o Abel vs. O Abel é mais alto que o Rui
    - o MaisAlto(rui, abel) vs. MaisAlto(abel, rui)
- Em **lógica de 1ª ordem**, os argumentos de predicados são funções ou constantes, mas **NUNCA** outros predicados
  - O Rui pensa que o Jorge conhece a Sara
    - o Pensa(rui, Conhece(jorge,sara)) - **Não** é fórmula de 1ª ordem

# A Linguagem da Aritmética

---

## Assinatura:

- Constantes: Números Naturais
  - $SF_0 = \{0, 1, 2, 3, \dots\}$
- Funções : Operações aritméticas
  - Unárias  $SF_1 = \{-, +, !\}$
  - Binárias  $SF_2 = \{+, -, *, /, ^, \text{mod}\}$
- Predicados: Relações de ordem
  - Binárias  $SP_2 = \{=, \neq, <, >, \leq, \geq, \}$
- Exemplos:
  - $\geq (4, 3)$                       cuja forma infixa é mais comum  $4 \geq 3$
  - $>(+ (4, 1), 6)$                       cuja forma infixa é mais comum  $(4 + 1) > 6$
  - $=(+ (! (5)), *(5, ! (4)))$               ou usando as forma infixa, prefixa e posfixa  $+5! = 5 * 4!$

# A Linguagem da Teoria de Conjuntos

---

## Assinatura:

- Constantes: Arbitrárias – vamos assumir apenas os números naturais e letras minúsculas para denotarem conjuntos
  - $SF_0 = \{0, 1, 2, 3, \dots\} \cup \{a, b, c, d\}$
- Predicados: Relações de pertença e de igualdade
  - Binárias  $SP_2 = \{=, \in\}$
- **Nota:** Há mais predicados (inclusão) mas não são necessários, podem ser definidos.
- Exemplos:
  - $1 \in a$            $2 \in a$            $1 \in b$            $2 \in b$            $3 \in b$
  - $a = b$            $1 = a$

# A Linguagem Tarski (Mundo dos Blocos)

---

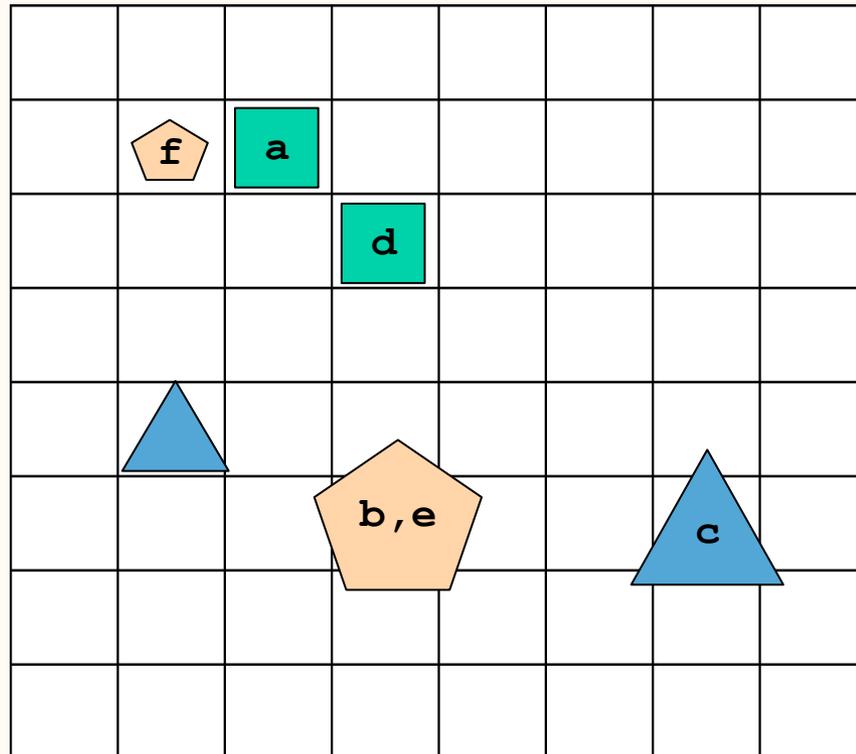
## Assinatura:

- Constantes: nomes que se podem dar a blocos
  - $SF_0 = \{a,b,c,d, \dots\}$
- Predicados Unários: Propriedades de tamanho e tipo dos blocos.
  - $SP_1 = \{ \text{Cube, Tet, Dodec, Small, Medium, Large} \}$
- Predicados Binários: Igualdade e Relações de tamanho, tipo, posição entre blocos
  - $SP_2 = \{ =, \text{Larger, Smaller, SameSize, SameShape, FrontOf, BackOf, SameRow, LeftOf, RightOf, SameCol, Adjoins} \}$
- Predicados Ternários: Bloco entre 2 blocos (os 3 blocos alinhados)
  - $SP_3 = \{ \text{Between} \},$

# Exemplos

- Tet(c)
- Dodec(a)
- Large(b)
- Medium(d)
- Larger(a,c)
- SameShape(a,d)
- SameSize(c,f)
- FrontOf(c,a)
- SameRow(b,c)
- LeftOf(b,c)
- SameCol(b,c)
- Adjoins(a,f)
- Adjoins(a,d)
- Between(d,a,c)
- Between(d,c,a)
- Between(a,d,c)
- $b = e$
- $a = d$

## Um Mundo de Blocos (em 2D)



# Aplicações à Informática

---

- Bases de Dados
  - Modelação
- Inteligência Artificial
  - Representação do Conhecimento
  - Raciocínio Automático
- Linguagens (e Modelos) de Programação
  - Semântica das Linguagens
  - Verificação de Propriedades de Programas
  - Linguagens de Programação em Lógica

# Avaliação de Conhecimentos

---

Página da Cadeira: [ssdi.di.fct.unl.pt/lc](http://ssdi.di.fct.unl.pt/lc)

## Avaliação Contínua

- 4 mini-testes (todos com peso 25% - duração 1 hora)
  - 14 de Outubro<sup>†</sup>: capítulos 1 – 4      Datas sujeitas a alterações
  - 11 de Novembro<sup>†</sup>: capítulos 5 – 9
  - 30 de Novembro: capítulos 9 -13
  - 16 de Dezembro<sup>†</sup>: capítulos 15 – 18 (partes)

<sup>†</sup> Datas sujeitas a alterações

## - Exame de Recurso

- Data e Hora a anunciar
- Não requer equivalência