

# Lógica Computacional

---

Argumentos válidos e sólidos

Métodos de Demonstração

Demonstrações formais

Regras de Inferência – Igualdade

Não-consequências lógicas

# Argumentos

---

## Exemplo:

1. Todos as aves voam.
  2. O Pingu é uma ave
  3. Logo o Pingu voa.
- Um argumento é uma sequência de frases das quais uma é justificada pelas outras.
  - A frase justificada é chamada a **conclusão**.
  - As frases que a justificam são chamadas **premissas**
  - Normalmente a conclusão é a última frase (“logo ...”) mas não necessariamente.

## Exemplo:

1. O Batman não voa.
  2. De facto, o Batman é um morcego.
  3. Os morcegos são mamíferos.
  4. E os mamíferos não voam
- Neste caso a conclusão é a primeira frase que se justifica (“de facto ...”) pelas premissas que se seguem.

# Validade de Argumentos - Consequência Lógica

---

- Um argumento é **válido** se a conclusão é verdadeira **sempre** que as premissas forem verdadeiras.
- Nesse caso diz-se que a conclusão é uma **consequência lógica** das premissas.

## Exemplo:

1. Todos os cães voam.
  2. O Snoopy é um cão.
  3. Logo o Snoopy voa.
- Para que um argumento seja válido não é necessário que a sua conclusão seja verdadeira!

## Exemplo:

1. Nenhuma ave voa.
  2. O Piu-Piu é uma ave.
  3. Logo o Piu-Piu voa.
- Um argumento pode não ser válido apesar de a sua conclusão ser verdadeira.

# Solidez de Argumentos

---

- Por si só, **a validade de um argumento não garante a verdade da conclusão**. Esta argumentação pode ser usada para avaliar a consequência de certas acções ou condições que ainda não se realizaram para avaliar se a conclusão é conveniente.

## Exemplo:

1. Quem compra algo tem de pagar.
  2. Vou comprar este carro
  3. Logo vou ter de pagar.
- Normalmente estamos interessados em argumentos **sólidos**, isto é que sejam não apenas válidos mas em que as premissas sejam verdadeiras. Nesse caso, a conclusão será necessariamente verdadeira!

## Exemplo:

1. Todos as aves respiram.
2. O Piu-Piu é uma ave.
3. Logo o Piu-Piu respira.

# Raciocínio e Demonstrações

---

- Normalmente uma conclusão (consequência) lógica não se obtém imediatamente a partir das premissas mas requer um conjunto de frases intermédias a caminho da conclusão.

## Exemplo:

1. Se um animal não respira morre.

2. A respiração requer ar.

Assume-se respiração por pulmões

3. Não há ar dentro de água.

4. O Snoopy é um animal.

5. Todos os cães ladram

6. Dentro de água não se pode respirar

consequência lógica de 2 e 3

7. Se o Snoopy não respira morre

consequência lógica de 1 e 4

8. O Snoopy morre dentro de água

consequência lógica de 6 e 7

- As frases 6, 7 e 8 são *todas* consequências lógicas das premissas, embora 6 e 7 tenham sido usados como passos intermédios para a conclusão 8 (principal).
- **Nota:** Nem todas as premissas são necessárias para validar um argumento. É o caso de 5.

# Sistemas e Demonstrações Formais

---

- Uma **demonstração formal**, ou **dedução**, é uma sequência de fórmulas da linguagem formal utilizada, em que as fórmulas correspondentes às premissas são identificadas inicialmente, e em que a conclusão é a última fórmula.
- Todos os passos (conclusões) intermédias têm de ser justificadas a partir de regras de inferência claras e geralmente num número limitado.
- As regras (e a linguagem) utilizadas definem um **sistema formal**.
- Nesta cadeira exploraremos dois sistemas formais: **Dedução Natural (DN)** e **Resolução (R)**.
- O sistema de dedução natural, como o nome indica, pretende simular o raciocínio humano, usando uma linguagem e um conjunto de regras de inferência próximas das usadas em linguagem natural.
- O sistema de resolução utiliza um subconjunto limitado dessa linguagem e menos regras de inferência, permitindo o mais fácil tratamento computacional.
- Apesar das suas diferenças ambos os sistemas têm o mesmo *poder de expressão* e *poder de demonstração*.

# Demonstrações Formais

---

- Em geral apresentaremos uma demonstração através de uma sequência de fórmulas (representando frases) ao longo de uma barra vertical, sendo as premissas separadas das conclusões (parciais e final) por uma barra horizontal.

## Exemplo:

1. <b>Cube (a)</b>	% <b>a</b> é um cubo
2. <b>a = b</b>	% <b>a</b> e <b>b</b> são o mesmo objecto
<hr/>	
3. <b>Cube (b)</b>	<b>Regra (?) 1, 2</b>

- Normalmente todas as fórmulas são numeradas, para identificar a sua ordem na sequência e para poderem ser referidas nos passos de inferência.
- Todas as fórmulas abaixo da barra horizontal (conclusões) devem ser justificadas por uma regra de inferência.

# Regras de Inferência

---

- No desenho de um sistemas formal há que escolher as suas regras de inferência.
- Elas devem ser simples, para ser simples determinar quando se aplicam, e devem constituir um conjunto suficientemente rico para podermos justificar as inferências desejadas.
- Além disso deverão ser universais, podendo ser aplicadas a qualquer linguagem, sem ser necessário entender o “significado” das fórmulas utilizadas.
- **Exemplo:**

1. <b>Cube (a)</b>	
2. <b>Cube (b)</b>	
<hr/>	
3. <b>SameShape (a ,b) ? 1 ,2</b>	

1. <b>C (a)</b>	
2. <b>C (b)</b>	
<hr/>	
3. <b>S (a ,b)                    ?? 1 ,2    ??</b>	

- Esta regra depende do conhecimento que se tenha do Mundo dos Blocos.
- Como se justificaria este argumento se os predicados fossem *renomeados*?

# Regras de Inferência - Igualdade

- Um predicado muito especial é o predicado de igualdade, já que ele aparece em todas as linguagens para indicar que um objecto pode ter 2 nomes.
- Assim sendo podem definir-se duas regras de inferência muito simples:
- **Introdução da igualdade:**
  - Um objecto é igual a si próprio.
- **Eliminação da Igualdade:**
  - Se um objecto tem dois nomes as propriedades que se atribuem a um dos nomes também se atribuem ao outro nome.
- Formalmente:

```
...  
k. a = a      Int =  
...
```

```
...  
k1.  $\phi[a]$       Nota:  
...  
k2. a = b      k > k1  
...  
k.  $\phi[a/b]$      k > k2  
...  
Elim =: k1, k2
```

# Regras de Inferência - Igualdade

---

- Estas regras permitem fazer algumas inferências básicas.

## Exemplo: Introdução da igualdade:

- Se **a** e **b** são o mesmo objecto e se **a** é um cubo, **b** também o é!

1. <b>Cube (a)</b>	
2. <b>a = b</b>	
<hr/>	
3. <b>Cube (b)</b>	<b>Elim =: 1,2</b>

- No entanto, a regra de eliminação da igualdade apenas permite substituir o nome da esquerda pelo da direita **e não o contrário**, Assim não permite concluir que:
  - Se **b** e **a** são o mesmo objecto e se **a** é um cubo, **b** também o é!

1. <b>Cube (a)</b>	
2. <b>b = a</b>	
<hr/>	
3. <b>Cube (b)</b>	<b>???</b>

# Regras de Inferência - Igualdade

---

- O problema aqui é que o sistema não garante, à partida, que se  $\mathbf{a} = \mathbf{b}$ , então também  $\mathbf{b} = \mathbf{a}$ , o que parece óbvio já que ambas as igualdades expressam o facto que  $\mathbf{a}$  e  $\mathbf{b}$  são nomes para o mesmo objecto.
- Em geral, essa inferência se se pode fazer para o predicado de igualdade, não se pode fazer para um qualquer predicado. De facto, de  $\mathbf{P(a,b)}$  não se pode concluir que  $\mathbf{P(b,a)}$ .
- No entanto, e como é bem conhecido, a igualdade goza não só da propriedade **reflexiva** (captada na regra da introdução da Igualdade) como ainda das propriedades **simétrica** e **transitiva**.
- Essas propriedades não são “primitivas” do sistema mas podem ser deduzidas no sistema formal.
- **Nota:** A outra propriedade importante da igualdade, a **substitutividade**, é a propriedade que o sistema formal incorpora na regra da Eliminação da Igualdade.

# Regras de Inferência - Igualdade

---

- Podemos deduzir a **Simetria da Igualdade** com a demonstração seguinte:

1. $a = b$	
<hr/>	
2. $a = a$	<b>Intro =</b>
3. $b = a$	<b>Elim =: 1,2</b>

- De notar que na aplicação da regra de Eliminação da Igualdade substituímos o símbolo **a** que aparece do lado esquerdo da fórmula 2, pelo símbolo **b**, tendo em atenção a igualdade anterior **a = b**.
- Assim sendo, já se pode fazer a demonstração anteriormente não permitida, interpondo a dedução da simetria!

1. <b>Cube (a)</b>	
2. $b = a$	
<hr/>	
3. $b = b$	<b>Intro =</b>
4. $a = b$	<b>Elim =: 2,3</b>
5. <b>Cube (b)</b>	<b>Elim =: 1,4</b>

# Regras de Inferência - Igualdade

---

- Em geral, e para evitar que as demonstrações se tornem demasiado longas, poderemos omitir a dedução da simetria da Igualdade e simplificar a demonstração da forma “intuitiva”

1. <b>Cube (a)</b>	
2. <b>b = a</b>	
<hr/>	
3. <b>Cube (b)</b>	<b>Elim =: 1,2</b>

- Uma outra convenção que tem sido feita na regra de Eliminação da Igualdade é *referir* as fórmulas utilizadas em ordem crescente. Em rigor a regra impõe que a primeira fórmula seja a fórmula da igualdade e a segunda a fórmula em que se substituem os símbolos.
- Num programa que implemente o sistema estes pequenos pormenores poderão ter alguma importância, mas usaremos sempre esta simplificação. A única condição que colocaremos é que as fórmulas usadas numa regra apareçam antes da fórmula que se obtem por aplicação da regra!

# Regras de Inferência - Igualdade

---

- A propósito, esta simplificação pode ser “justificada” através de uma “regra” implícita em qualquer sistema de dedução, que permite repetir (reiterar) uma fórmula já conhecida, já que uma fórmula verdadeira é sempre verdadeira.

1. <b>Cube (b)</b>	
2. <b>b = a</b>	
<hr/>	
3. <b>Cube (b)</b>	<b>Reit: 1</b>
4. <b>Cube (a)</b>	<b>Elim =: 2,3</b>

- Sem mais “complicações” podemos agora demonstrar a transitividade da igualdade:

1. <b>a = b</b>	
2. <b>b = c</b>	
<hr/>	
3. <b>a = c</b>	<b>Elim =: 1,2</b>

# Consequência Lógica - Possível Demonstrar?

---

- Com um sistema formal pretende-se deduzir consequências lógicas através de demonstrações que sigam um conjunto de regras de inferência do sistema.
- Será que isso é sempre possível? Se uma fórmula é uma consequência lógica das premissas será que é sempre possível obter uma demonstração com as regras do sistema?
- Este é o problema da **completude**. Neste momento a linguagem que temos utilizado apenas permite formular proposições simples e as únicas regras de inferência são a Introdução e Eliminação da Igualdade, pelo que é cedo para abordarmos esta questão.
- Em antecipação, digamos que existem linguagens, com poder de expressão para as quais existem sistemas formais completos. É o que estudaremos inicialmente com a Lógica Proposicional.
- Para sistemas com maior poder de expressão, nomeadamente aqueles que possam especificar a aritmética a resposta é **não**. Este é o enunciado do famoso **teorema da Incompletude de Godel**.

# Não-Consequência Lógica - Contraexemplos

---

- Mesmo com sistemas completos coloca-se a questão de saber como se “deduz” que uma fórmula não é uma consequência lógica das premissas.
- Claro está que podemos tentar testar todas as possíveis combinações de regras de inferência sobre todas as premissas e fórmulas intermédias mas esse processo é potencialmente infinito.
- Assim sendo a melhor forma de provar a **não**-consequência é através da definição: Determinar uma situação em que as premissas sejam verdadeiras e a conclusão seja falsa.

1. Cube (a)	
2. Large (b)	
<hr/>	
3. Cube (b)	???

1. Cube (a)	
2. SameShape (a, b)	
<hr/>	
3. Tet (b)	???

- De notar que provar a não consequência lógica não é a mesma coisa que provar a consequência lógica da fórmula negada. Por exemplo, **não concluir que b é um cubo** não é o mesmo que **concluir que b não é um cubo** (pode ser ou não)

# Não Consequência Lógica - Contraexemplo

1. **Cube (a)**

2. **Large (b)**

3. **Cube (b) ???**

1. **Cube (a)**

2. **SameShape (a, b)**

3. **Tet (b) ???**

