Universidade do Minho
Escola de Engenharia
Departamento de Informática

Programação em Lógica

LICENCIATURA EM ENGENHARIA INFORMÁTICA
MESTRADO integrado EM ENGENHARIA INFORMÁTICA
Inteligência Artificial
2022/23

ISLab
Synthetic Intelligence Lab

A Lógica
Teoria da Prova

- Axiomas: conjunto inicial de fórmulas lógicas
- Teoremas: fórmulas derivadas a partir dos axiomas e/ou teoremas (consequências semânticas)
- Regras de Inferência: conjunto de regras de derivação
  - *Modus ponens* $\{ (A \text{ se } B), B \} \vdash A$ (*sound* – válida)
  - *Modus tollens* $\{ (A \text{ se } B), \neg A \} \vdash \neg B$ (*sound* – válida)
  - *Modus mistakens* $\{ (A \text{ se } B), A \} \vdash B$ (*unsound* – não válida)
- Sistema de Inferência: união dos axiomas e das regras de derivação $\mathcal{R}$
- Prova: sequência $<s_1, s_2, ...>$ de s_i que são axiomas ou são derivações usando $\mathcal{R}$ e um subconjunto dos membros da sequência que precedem s_i;
  - A sequência é uma prova para s_n (derivação ou dedução)
- Teoria: união dos axiomas e de todos os teoremas derivados usando $\mathcal{R}$
  - Diz-se consistente sse não existe nenhuma fórmula s tal que, na teoria $\mathcal{T}$, exista s e $\neg s$
- **Nenhuma destas considerações toma em linha de conta o significado!**
  Apenas a estrutura sintática!!!

ISLab
Synthetic Intelligence Lab

A Programação em Lógica

- A Programação em Lógica é um formalismo computacional que combina 2 princípios básicos:

1. Usa a Lógica para representar conhecimento
   (representação de pressupostos e de conclusões)

2. Usa a Inferência para manipular o conhecimento
   (estabelecer as relações lógicas entre os pressupostos e as conclusões)
   (mecanizar os procedimentos de prova; raciocinar)

ISLab
Synthetic Intelligence Lab

Caracterização da Programação em Lógica PROLOG

- Um programa em PROLOG é criado pela adição de fórmulas designadas por **cláusulas**
- As cláusulas podem ser de 3 tipos:
  - Factos: expressam algo que é sempre verdadeiro

    p. `filho( xico,quim ).`

  - Regras: expressam algo que é verdadeiro, dependente da veracidade das condições

    p se q. `pai( josé,joão ) se filho( joão,josé ).`

  - Questões: expressam algo que é verdadeiro, dependente da veracidade das condições

    ?q. `? pai( josé,joão ).`

    ¬q. `¬pai( josé,joão ).`

ISLab
Synthetic Intelligence Lab

Caracterização da Programação em Lógica PROLOG

- Um programa em PROLOG é criado pela adição de fórmulas designadas por **cláusulas**
- As cláusulas podem ser de 3 tipos:
  - Factos: expressam algo que é sempre verdadeiro

    p. `filho( xico,quim ).`

  - Regras: expressam algo que é verdadeiro, dependente da veracidade das condições

    p se q. `pai( josé,joão ) se filho( joão,josé ).`

  - Questões: expressam algo que é verdadeiro, dependente da veracidade das condições

    ?q. `? pai( josé,joão ).`

    ¬q. `¬pai( josé,joão ).`

p se q.

Cláusulas de Horn

- Clausulado de Horn (notação clausal da Lógica de primeira ordem)
  - É uma versão restrita do Cálculo Predicativo
  - É uma formula bem formada
  - Todas as fórmulas estão quantificadas universalmente
  - Todas as fórmulas são fechadas
  - As fórmulas lógicas admitem, apenas, 1 termo na disjunção positiva de literais

Cláusulas de Horn

  - $\neg q_i \vee p_j$
  - em que $i \geq 0$

    $0 \leq j \leq 1$

ISLab
Synthetic Intelligence Lab

Definições

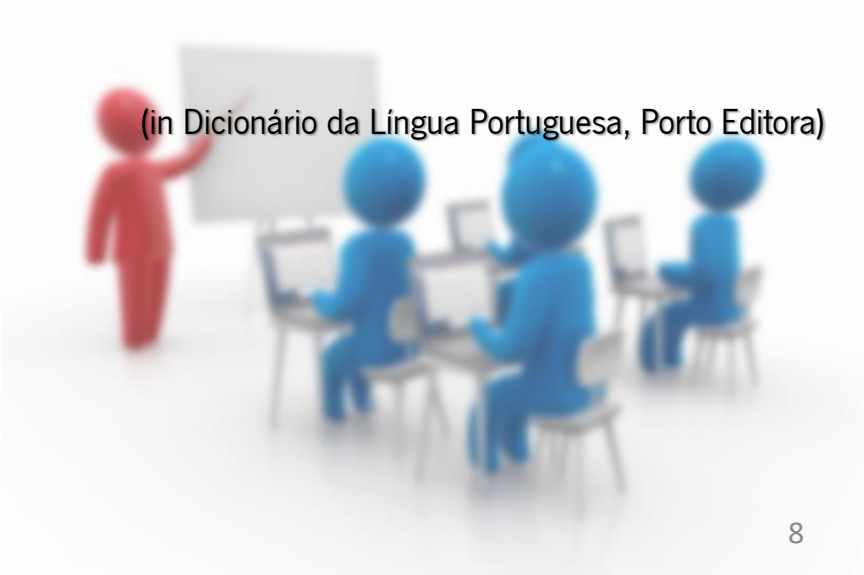
- Raciocínio:
  - Conjunto de pensamentos encadeados
  - Relacionamento de factos que levam a uma conclusão
  - Mistura dos 2 anteriores

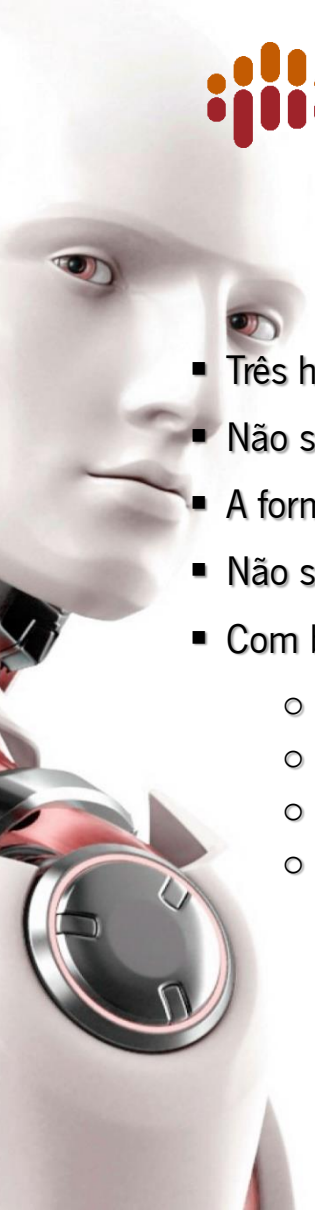
ISLab
Synthetic Intelligence Lab

Definições

- Raciocínio:
  1. ...
  2. Encadeamento lógico de pensamentos
  3. ...
  4. [LÓGICA] Operação discursiva do pensamento mediante a qual concluímos que uma ou várias proposições (premissas) implicam a verdade (...) de uma outra proposição (conclusão)

(in Dicionário da Língua Portuguesa, Porto Editora)





ISLab
Synthetic Intelligence Lab

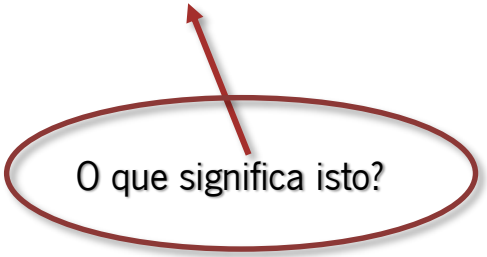
Raciocínio

- Três homens, António, Belmiro e Carlos, são cônjuges de Dulce, Eduarda e Filipa.
- Não se sabe quem é casado com quem.
- A formação deles é em Engenharia, Advocacia e Medicina.
- Não se sabe quem faz o quê.
- Com base nos dados abaixo, descubra o nome de cada esposa e a profissão de cada homem:
  - O médico é casado com Filipa;
  - Carlos é advogado;
  - Eduarda não é casada com Carlos;
  - Belmiro não é médico.

(adaptado de: www.novaconcursos.com.br/portal/dicas/questao-raciocinio-logico-para-concursos)

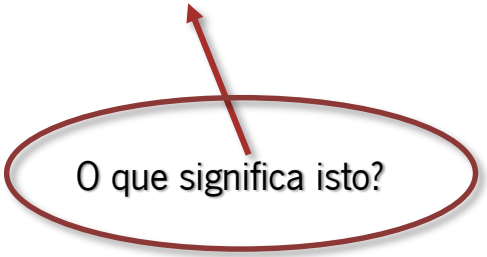
Raciocínio e Inferência

- Para convencer de que a resposta não é “um tiro de sorte”, é necessário expor as razões que levam a atingir a conclusão!



Raciocínio e Inferência

- Para convencer de que a resposta não é "um tiro de sorte", é necessário expor as razões que levam a atingir a conclusão!



- Para que a conclusão seja aceite, torna-se necessário explicar o mecanismo de raciocínio aplicado, ou seja, o **processo de inferência**.

ISLab
Synthetic Intelligence Lab

Definições

- Inferência:
  - Diz-se do processo aplicado, que permite passar das premissas à conclusão.
  - Inferência: dedução ou conclusão.

(in Dicionário Priberam da Língua Portuguesa)

ISLab
Synthetic Intelligence Lab

Definições

- Inferência:
  - Diz-se do processo aplicado, que permite passar das premissas à conclusão.
  - Inferência: dedução ou conclusão.

(in Dicionário Priberam da Língua Portuguesa)

- Regras de inferência:
  - *Modus ponens*:
    - de: P , $P \rightarrow Q$
    - infere-se: Q

ISLab
Synthetic Intelligence Lab

Definições

- Inferência:
  - Diz-se do processo aplicado, que permite passar das premissas à conclusão.
  - Inferência: dedução ou conclusão.

(in Dicionário Priberam da Língua Portuguesa)

- Regras de inferência:
  - *Modus ponens*: (*modus ponendo ponens Latim significa "a maneira que afirma afirmando")*
    - de: P , $P \rightarrow Q$
    - infere-se: Q

P	Q	P → Q
𝕍	𝕍	𝕍
𝕍	𝔽	𝔽
𝔽	𝕍	𝕍
𝔽	𝔽	𝕍

ISLab
Synthetic Intelligence Lab

Definições

- Inferência:
  - Diz-se do processo aplicado, que permite passar das premissas à conclusão.
  - Inferência: dedução ou conclusão.

(in Dicionário Priberam da Língua Portuguesa)

- Regras de inferência:
  - *Modus ponens*:
    - de: $P , P \rightarrow Q$
    - infere-se: Q
  - *Modus tollens*:
    - de: $P \rightarrow Q , \neg Q$
    - infere-se: $\neg P$

Definições

- Inferência:
  - Diz-se do processo aplicado, que permite passar das premissas à conclusão.
  - Inferência: dedução ou conclusão.

(in Dicionário Priberam da Língua Portuguesa)

- Regras de inferência:
  - *Modus ponens*:
    - de: P , $P \rightarrow Q$
    - infere-se: Q
  - *Modus tollens* (Latim: modo que nega por negação):
    - de: $P \rightarrow Q$, $\neg Q$
    - infere-se: $\neg P$

P	Q	¬Q	P → Q
𝕍	𝕍	𝔽	𝕍
𝕍	𝔽	𝕍	𝔽
𝔽	𝕍	𝔽	𝕍
𝔽	𝔽	𝕍	𝕍

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução

- A regra de inferência *modus ponens* permite derivar como verdadeira a conclusão de uma cláusula pela prova das respetivas condições:

$$\{ (A \text{ se } B), B \} \vdash A$$

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução

- A regra de inferência *modus ponens* permite derivar como verdadeira a conclusão de uma cláusula pela prova das respetivas condições:

$$\{ (A \text{ se } B), B \} \vdash A$$

- A aplicação da regra de inferência *modus tollens* permite dirigir a procura da prova para um ponto concreto do processo de raciocínio:

$$\{ (A \text{ se } B), \neg A \} \vdash \neg B$$

  - o que permite desenvolver um **mecanismo de prova por contradição**.

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução

- Supor que temos um programa $\mathcal{P}$, no qual queremos determinar a derivabilidade de uma questão A:
  - i. Admita-se a negação de A:
    $\neg A$
  - ii. Insira-se a negação de A no programa $\mathcal{P}$:
    $\mathcal{P} \cup \neg A$
- Se acontecer gerar-se uma contradição
  $\{ A, \neg A \} \equiv \square$
  tal significa que a questão inicial A é derivável de $\mathcal{P}$.

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução

- Supor que temos um programa $\mathcal{P}$, no qual queremos determinar a derivabilidade de uma questão A:
  - i. Admita-se a negação de A:
    $\neg A$
  - ii. Insira-se a negação de A no programa $\mathcal{P}$:
    $\mathcal{P} \cup \neg A$
- Se acontecer gerar-se uma contradição
  $\{ A, \neg A \} \equiv \square$
  tal significa que a questão inicial A é derivável de $\mathcal{P}$.

- Regra de inferência modus tollens, com B = verdadeiro
  $\{ (A \text{ se } B), \neg A \} \vdash \neg B$
  $\{ (A \text{ se } \mathbb{V}), \neg A \} \vdash \neg \mathbb{V}$
  $\{ A, \neg A \} \vdash \mathbb{F}$
  $\{ A, \neg A \} \vdash \square$

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Considere-se o programa $\mathcal{P}$ indicado ao lado

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (I)

- O João é filho do José?

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Admita-se a colocação da questão escrita acima

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (I)

- O João é filho do José?

  ¬filho(joao,jose)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Em termos lógicos, a questão é descrita na formulação indicada

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (I)

- O João é filho do José?

  ¬filho(joao,jose)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

De todas as cláusulas do programa $\mathcal{P}$, apenas as que oferecem conclusões sobre o predicado `filho/2` contribuirão para o desenvolvimento a prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (I)

- O João é filho do José?

  ¬filho(joao,jose)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

De entre as três cláusulas assinaladas, a primeira será utilizada no primeiro passo do desenvolvimento da (árvore de) prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (I)

- O João é filho do José?

¬filho(joao,jose)

joao | joao, jose | jose

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

O termo joao unifica com joao; o termo jose unifica com jose;

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (I)

- O João é filho do José?

¬filho(joao,jose)

joao | joao, jose | jose

¬(verdade)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Nas condições estabelecidas pelas unificações, que utilizaram a primeira cláusula de $\mathcal{P}$, a questão inicial é reduzida à atual, por se tratar de um facto

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (I)

- O João é filho do José?

¬filho(joao,jose)

joao| joao, jose | jose

¬(verdade)

-|-

□

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

De onde se retira uma contradição □ (independentemente de quaisquer unificações)

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Pretende-se saber se o José é pai do João

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

  ¬pai(jose,joao)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Em termos lógicos, a questão é descrita na formulação indicada

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

  ¬pai(jose,joao)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

De todas as cláusulas do programa $\mathcal{P}$, apenas as que oferecem conclusões sobre o predicado `pai/2` contribuirão para o desenvolvimento a prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

  ¬pai(jose,joao)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

A única cláusula do predicado `pai/2` assinalada será utilizada no primeiro passo do desenvolvimento da (árvore de) prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

¬pai(jose,joao)

jose | P, joao | F

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

O termo jose unifica com a variável P; o termo joao unifica com a variável F;

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

¬pai(jose,joao)
jose | P, joao | F
¬filho(joao,jose)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Para dar continuidade à (árvore de) prova, reduz-se a questão inicial à prova das condições da cláusula usada para proceder às unificações

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

¬pai(jose,joao)
jose | P, joao | F
¬filho(joao,jose)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

De todas as cláusulas do programa $\mathcal{P}$, apenas as que oferecem conclusões sobre o predicado `filho/2` contribuirão para o desenvolvimento a prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

¬pai(jose,joao)

jose | P, joao | F

¬filho(joao,jose)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

De entre as três cláusulas assinaladas, a primeira será utilizada no primeiro passo do desenvolvimento da (árvore de) prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

¬pai(jose,joao)

jose | P, joao | F

¬filho(joao,jose)

joao | joao, jose | jose

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

O termo joao unifica com joao; o termo jose unifica com jose;

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

¬pai(jose,joao)
- jose | P, joao | F

¬filho(joao,jose)
- joao | joao, jose | jose

¬(verdade)

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Nas condições estabelecidas pelas unificações, que utilizaram a primeira cláusula de $\mathcal{P}$, a questão anterior é reduzida à atual, por se tratar de um facto

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (II)

- O José é pai do João?

¬pai(jose,joao)
└── jose| P, joao | F
¬filho(joao,jose)
└── joao| joao, jose | jose
¬(verdade)
└── -|-
□

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

De onde se retira uma contradição □ (independentemente de quaisquer unificações)

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Admita-se a colocação da questão escrita acima

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?

  ¬filho(jose,joao)

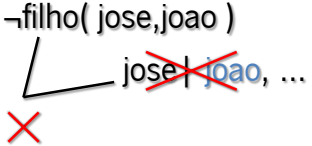
```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Em termos lógicos, a questão é descrita na formulação indicada; de entre as três cláusulas do predicado `filho/2`, a primeira será utilizada para proceder ao desenvolvimento da (árvore de) prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?



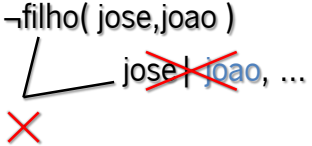

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

O termo jose não unifica com joao por se tratarem de duas constantes diferentes, pelo que o procedimento de prova não pode evoluir através deste ramo de prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?



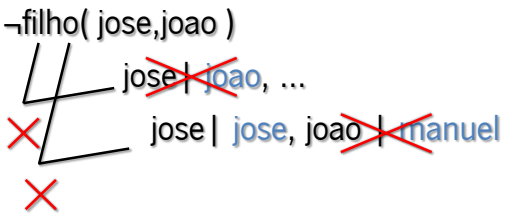

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Abandonando a primeira cláusula, toma-se a segunda como alternativa para dar continuidade ao desenvolvimento da (árvore de) prova;

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?



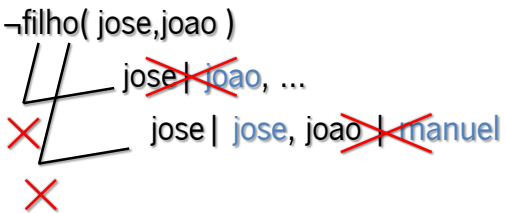

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

O termos jose unifica com jose; o termo joao não unifica com manuel por se tratarem de duas constantes diferentes, pelo que o procedimento de prova não pode evoluir através deste ramo de prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?



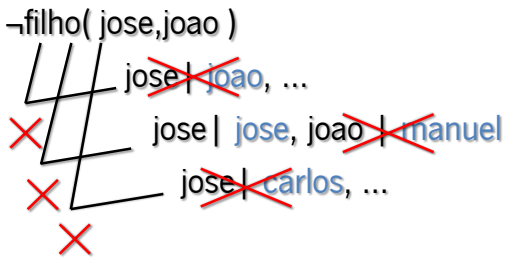

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Abandonando, também, a segunda cláusula, toma-se a terceira (e última) como alternativa para dar continuidade ao desenvolvimento da (árvore de) prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?



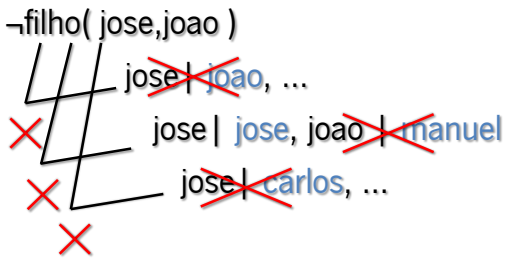

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

O termo jose não unifica com carlos por se tratarem de duas constantes diferentes, pelo que o procedimento de prova não pode evoluir através deste ramo de prova

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?



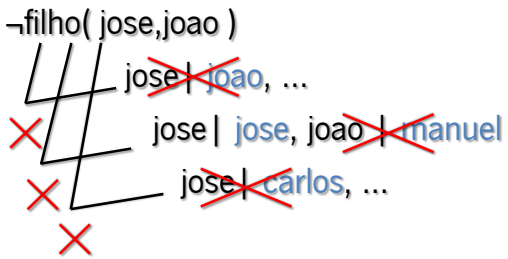

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Não havendo mais cláusulas no predicado filho/2 que possam ser consideradas mais alternativas na (árvore de) prova, o procedimento de aplicação do algoritmo de resolução termina sem alcançar qualquer contradição

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (III)

- O José é filho do João?




```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

~~O José é filho do João?~~

Tal significa que a questão inicial não gera nenhuma contradição em $\mathcal{P}$, logo, é falsa.

ISLab
Synthetic Intelligence Lab

Algoritmo de Resolução
Exemplo de aplicação (IV)

¬ filho(X,jose)

Qual é o significado desta questão

Desenvolver a árvore de prova para esta questão

```
% filho: Filho,Pai → {𝕍,𝔽}

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

% pai: Pai,Filho → {𝕍,𝔽}

pai( P,F ) :- filho( F,P ).
```

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Programação em Lógica

LICENCIATURA EM ENGENHARIA INFORMÁTICA
MESTRADO integrado EM ENGENHARIA INFORMÁTICA
Inteligência Artificial
2022/23