Agentes Lógicos

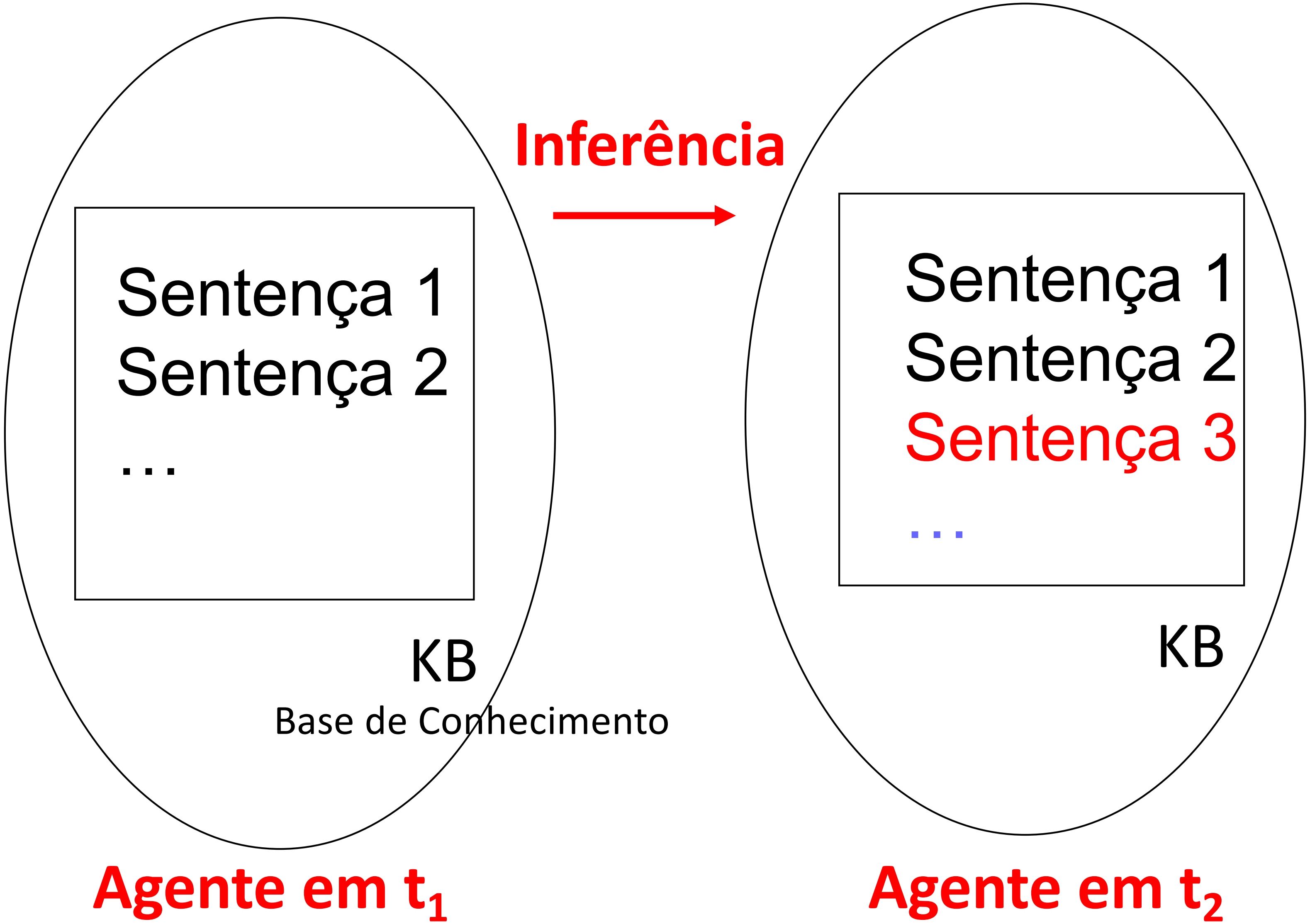
Inteligência Artificial
PCS3438

Anna Helena Reali Costa
Escola Politécnica da USP
Engenharia de Computação (PCS)

Agentes Baseados em Conhecimento

- Importante pois o agente pode combinar o resultado de percepções correntes com um conhecimento anterior.
- Um agente baseado em conhecimento é flexível
  - seu conhecimento pode ser alterado ou adicionado.

Agentes Baseados em Conhecimento



Lógica

- É uma "álgebra" para manipular somente dois valores: true (T) e false (F)

Lógica

- **Sintaxe**
  - Especificação das expressões legais da linguagem
- **Semântica**
  - Significado das expressões da linguagem
- **Inferência**
  - Regras que manipulam expressões da linguagem e que preservam alguns aspectos semânticos

Sintaxe

Formalização de uma linguagem lógica

Sintaxe

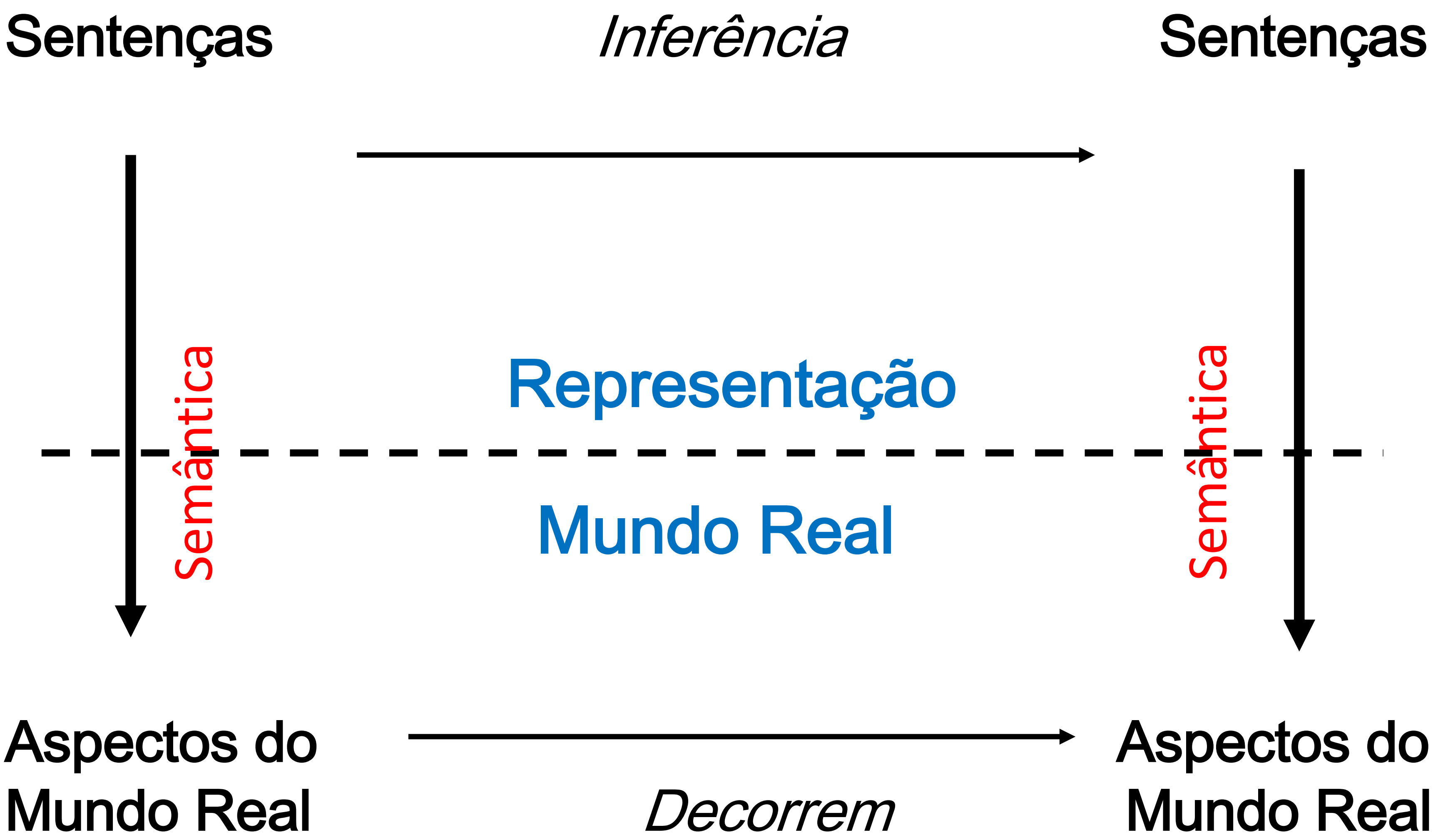
Especificação
das expressões
legais da linguagem

Ex: linguagem aritmética

α: x + y = 4 é uma fórmula bem formada (fbf)

β: x 4 y + = não é uma fbf

Lógica



Sentenças
Inferência
Sentenças
Semântica
Representação
Mundo Real
Semântica
Aspectos do Mundo Real
Decorrem
Aspectos do Mundo Real

LÓGICA PROPOSICIONAL

Sintaxe: Proposições

Trata-se do cálculo mais simples que existe.

Uma **proposição** é um enunciado declarativo.

- **p**: Todo imigrante italiano é palmeirense
- **q**: João gosta de comer feijoada

Proposições podem ser **verdadeiras** ou **falsas**.

Sintaxe: Conectivos Lógicos

Sentenças complexas podem ser construídas a partir de proposições simples, parênteses e de **conectivos lógicos**:

Mais alta

precedência

Mais baixa

- $\neg$: negação (não)
- $\wedge$: conjunção (e)
- $\vee$: disjunção (ou)
- $\rightarrow$: implicação (condicional, se...então)
- $\leftrightarrow$: bicondicional (se e somente se)

Sintaxe do Cálculo Proposicional em BNF

Símbolo inicial: <sentença>

<sentença>::= <sentença atômica> |
¬<sentença>|
(<sentença> <conectivo> <sentença>)

<sentença atômica>::= V | F | <símbolo>

<símbolo>::= p | q | r | s | ...

<conectivo>::= $\wedge$ | $\vee$ | $\rightarrow$ | $\leftrightarrow$

Semântica: Tabela Verdade

Caso uma fbf contenha n sentenças atômicas, existem 2^n interpretações distintas, que podem ser colocadas em uma **tabela da verdade.**

Conectivo NÃO (Negação)	
p	**$\neg$p**
F	V
V	F

Semântica: Tabela Verdade

Conectivo **OU** *Disjunção*			Conectivo **E** *Conjunção*		
p	**q**	**p ∨ q**	**p**	**q**	**p ∧ q**
F	**F**	**F**	**F**	**F**	**F**
F	**V**	**V**	**F**	**V**	**F**
V	**F**	**V**	**V**	**F**	**F**
V	**V**	**V**	**V**	**V**	**V**

Semântica: Tabela Verdade

Conectivo *Condicional*			Conectivo *Bicondicional*		
p	**q**	**$p \rightarrow q$**	**p**	**q**	**$p \leftrightarrow q$**
F	**F**	**V**	**F**	**F**	**V**
F	**V**	**V**	**F**	**V**	**F**
V	**F**	**F**	**V**	**F**	**F**
V	**V**	**V**	**V**	**V**	**V**

Semântica: Fórmulas Bem Formadas

Uma fórmula bem formada α é :

- **Válida (tautologia)** se sempre tiver valor **V**;
- **Satisfazível** se tiver algum valor **V**;
- **Inválida** se tiver algum valor **F**;
- **Insatisfazível (contradição)** se sempre tiver valor **F**;
- **Contingente** se não for válida nem insatisfazível.

Consequência Lógica: $\alpha \vDash \beta$

Sempre que α é V, β também é V.

Teorema da Dedução

Dadas as fbfs $\beta_1, \beta_2, \ldots, \beta_n$ e α,
α é consequência lógica de $\beta_1, \beta_2, \ldots, \beta_n$ sse a fbf $\beta_1 \wedge \beta_2 \wedge \ldots \wedge \beta_n \rightarrow \alpha$
for uma **tautologia**.

Teorema da Contradição

Dadas as fbfs $\beta_1, \beta_2, \ldots, \beta_n$ e α,
α é consequência lógica de $\beta_1, \beta_2, \ldots, \beta_n$
sse a fbf $\beta_1 \wedge \beta_2 \wedge \ldots \wedge \beta_n \wedge \neg \alpha$
for uma **contradição**.

Exemplo de Dedução

- Verificar se p é consequência lógica de

$$(p \vee h) \wedge \neg h$$

- Por dedução, basta mostrar que é **válida** a fbf:

$$(p \vee h) \wedge \neg h \rightarrow p$$

p	h	$p \vee h$	$(p \vee h) \wedge \neg h$	$(p \vee h) \wedge \neg h \rightarrow p$
F	F	F	F	V
F	V	V	F	V
V	F	V	V	V
V	V	V	F	V

TAUTOLOGIA!

Exemplo de Dedução

- Verificar se p é consequência lógica de

$$(p \vee h) \wedge \neg h$$

- Por contradição, basta mostrar que é **contradição** a fbf: $(p \vee h) \wedge \neg h \rightarrow p$

p	h	$p \vee h$	$(p \vee h) \wedge \neg h$	$\neg p$	$(p \vee h) \wedge \neg h \wedge \neg p$
F	F	F	F	V	F
F	V	V	F	V	F
V	F	V	V	F	F
V	V	V	F	F	F

CONTRADIÇÃO!

Consequência Lógica

Entretanto, procedimentos usando tabela verdade podem ser muito custosos!

Felizmente, existem propriedades s**intéticas** que podem auxiliar a verificação ou geração de uma consequência lógica: **regras de inferência**.

Regras de Inferência

Modus Ponens	$\alpha, \alpha \rightarrow \beta \models \beta$
Modus Tollens	$\alpha \rightarrow \beta, \neg\beta \models \neg\alpha$
Silogismo Hipotético	$\alpha \rightarrow \beta, \beta \rightarrow \gamma \models \alpha \rightarrow \gamma$
Silogismo Disjuntivo	$\alpha \vee \beta, \neg \alpha \models \beta$
Simplificação	$\alpha \wedge \beta \models \alpha$
Conjunção	$\alpha, \beta \models \alpha \wedge \beta$
Adição	$\alpha \models \alpha \vee \beta$
Contraposição	$\alpha \rightarrow \beta \models \neg\beta \rightarrow \neg\alpha$
Resolução	$\alpha \vee \beta, \neg \alpha \vee \gamma \models \beta \vee \gamma$

No caso do Modus Ponens, lê-se: Se α for verdade e também $\alpha \rightarrow \beta$ for verdade, então obrigatoriamente β será verdade.

Outras propriedades

$$\alpha \wedge \beta \equiv \beta \wedge \alpha \quad \text{"}Comutativa\text{"}$$

$$\alpha \vee \beta \equiv \beta \vee \alpha \quad \text{"}Comutativa\text{"}$$

$$(\alpha \wedge \beta) \wedge \delta \equiv \alpha \wedge (\beta \wedge \delta) \quad \text{"}Associativa\text{"}$$

$$(\alpha \vee \beta) \vee \delta \equiv \alpha \vee (\beta \vee \delta) \quad \text{"}Associativa\text{"}$$

$$\neg(\neg \alpha) \equiv \alpha \quad \text{"}Exclusão\,de\,dupla\,negação\text{"}$$

$$\alpha \to \beta \equiv \neg \alpha \vee \beta \quad \text{"}Exclusão\,da\,implicação\text{"}$$

$$\alpha \leftrightarrow \beta \equiv (\alpha \to \beta) \wedge (\beta \to \alpha) \quad \text{"}Exclusão\,da\,bicondicional\text{"}$$

$$\neg(\alpha \wedge \beta) \equiv \neg \alpha \vee \neg \beta \quad \text{"}De\,Morgan\text{"}$$

$$\neg(\alpha \vee \beta) \equiv \neg \alpha \wedge \neg \beta \quad \text{"}De\,Morgan\text{"}$$

$$\alpha \wedge (\beta \vee \delta) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \delta) \quad \text{"}Distributiva\text{"}$$

$$\alpha \vee (\beta \wedge \delta) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \delta) \quad \text{"}Distributiva\text{"}$$

Inferência por computador

- Inferência (ou raciocínio – *reasoning*) por computador nada mais é do que efetuar um processo de **busca**, tomando expressões lógicas e aplicando regras de inferência a elas
  - Qual expressão lógica usar?
  - Qual regra de inferência usar?

Demonstrações por Dedução

Prova Direta (também chamada de Dedução Lógica): Parte-se das premissas e chega-se à conclusão. Usa-se o **Teorema da Dedução**.

Prova por Contradição (também chamada de Refutação): Utilizando o **Teorema da Contradição**, nega-se a conclusão e prova-se que a conjunção das premissas com esta conclusão negada é uma contradição.

Demonstrações por Dedução

Exemplo: Verificar se o argumento lógico é válido por dedução lógica:

- Se as uvas caem, então a raposa as come
- Se a raposa as come, então estão maduras
- As uvas estão verdes ou caem

Logo

- A raposa come as uvas se e só se as uvas caem

Demonstração por Dedução

1. Se as uvas caem, então a raposa as come
2. Se a raposa as come, então estão maduras
3. As uvas estão verdes ou caem

Nomeando:

p: as uvas caem
q: a raposa come as uvas
r: as uvas estão maduras

1. $p \rightarrow q$
2. $q \rightarrow r$
3. $\neg r \vee p$

Demonstrações por Dedução Lógica

Nomeando:

p: as uvas caem

q: a raposa come as uvas

r: as uvas estão maduras

C1: $p \rightarrow q$ 1a. premissa

C2: $q \rightarrow r$ 2a. premissa

C3: $\neg r \vee p$ 3a. premissa

C4: $r \rightarrow p$ (Subst. Equiv. C3)

C5: $q \rightarrow p$ (Silog. Hip. C2 e C4)

C6: $(p \rightarrow q) \wedge (q \rightarrow p)$ (Conjunção C1 e C5)

C7: $p \leftrightarrow q$ (Subst. Equiv. C6) *cqd*

Demonstração por Contradição

Nomeando:

p: as uvas caem

q: a raposa come as uvas

r: as uvas estão maduras

C1: $p \rightarrow q$	1a. premissa
C2: $q \rightarrow r$	2a. premissa
C3: $\neg r \lor p$	3a. Premissa
C4: $\neg(p \leftrightarrow q)$	Negação da conclusão

Demonstrações por Resolução

- Muito utilizada pois além de **correta,** é **completa** para a demonstração por **refutação**
  - o computador só precisa escolher o par de expressões lógicas para **aplicar sempre a mesma regra**

- Para aplicar a regra da resolução, o conjunto de fbfs deve ser transformado numa <u>**conjunção de cláusulas**</u>

Uma cláusula é uma disjunção de literais

(átomos ou átomos negados)

Demonstrações utilizando Refutação por Resolução

Refutação:	p1:	p	(premissa)
	p2:	$\neg p \vee q$	(f. clausal premissa)
	p3:	$\neg q \vee r$	(f. clausal premissa)
	p4:	$\neg r$	(neg. da conclusão)
	C4:	q	(Resolução p1 e p2)
	C5:	r	(Resolução p3 e C4)
	C6:	$\{\}$	(Resolução p4 e C5)

Cláusulas

Passos para transformar fbf em cláusula:

1. $\alpha \leftrightarrow \beta$ substitui por $(\alpha \to \beta) \wedge (\beta \to \alpha)$
2. $\alpha \to \beta$ substitui por $\neg\alpha \vee \beta$
3. Trata negação: $\neg(\neg\alpha) \equiv \alpha$

$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$

4. $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

Prova: Resolução por Refutação

- Se as uvas caem, então a raposa as come
- Se a raposa as come, então estão maduras
- As uvas estão verdes ou caem

Logo: A raposa come as uvas se e só se as uvas came

C1: $p \rightarrow q$	1a. premissa
C2: $q \rightarrow r$	2a. premissa
C3: $\neg r \vee p$	3a. Premissa
C4: $\neg(p \leftrightarrow q)$	Negação da conclusão

C1: $\neg p \vee q$	1a. premissa
C2: $\neg q \vee r$	2a. premissa
C3: $\neg r \vee p$	3a. Premissa
C4: *prox. slide*	Negação da conclusão

Transformação em cláusula

$\neg(p \leftrightarrow q) \equiv \neg((p \rightarrow q) \wedge (q \rightarrow p)) \equiv$

$\equiv \neg((\neg p \vee q) \wedge (\neg q \vee p)) \equiv$

$\equiv \neg(\neg p \vee q) \vee \neg(\neg q \vee p) \equiv$

$\equiv (p \wedge \neg q) \vee (q \wedge \neg p) \equiv$

$\equiv ((p \wedge \neg q) \vee q) \wedge ((p \wedge \neg q) \vee \neg p) \equiv$

$\equiv (p \vee q) \wedge (q \vee \neg q) \wedge (p \vee \neg p) \wedge (\neg q \vee \neg p) \equiv$

$\equiv (p \vee q) \wedge (\neg q \vee \neg p)$

Prova: Resolução por Refutação

C1: $\neg p \vee q$	1a. premissa
C2: $\neg q \vee r$	2a. premissa
C3: $\neg r \vee p$	3a. Premissa
C4: $p \vee q$	Negação da conclusão (i)
C5: $\neg q \vee \neg p$	Negação da conclusão (ii)
C6: $p \vee r$	Res(C2, C4)
C7: p	Res(C6, C3)
C8: $\neg p$	Res(C1, C5)
C9: { }	Res(C7, C8) → contradição!

Conclusão da Prova

Portanto, é verdade que a raposa come as uvas se e só se as uvas caem, dadas as premissas:

- Se as uvas caem, então a raposa as come
- Se a raposa as come, então estão maduras
- As uvas estão verdes ou caem

Conclusão

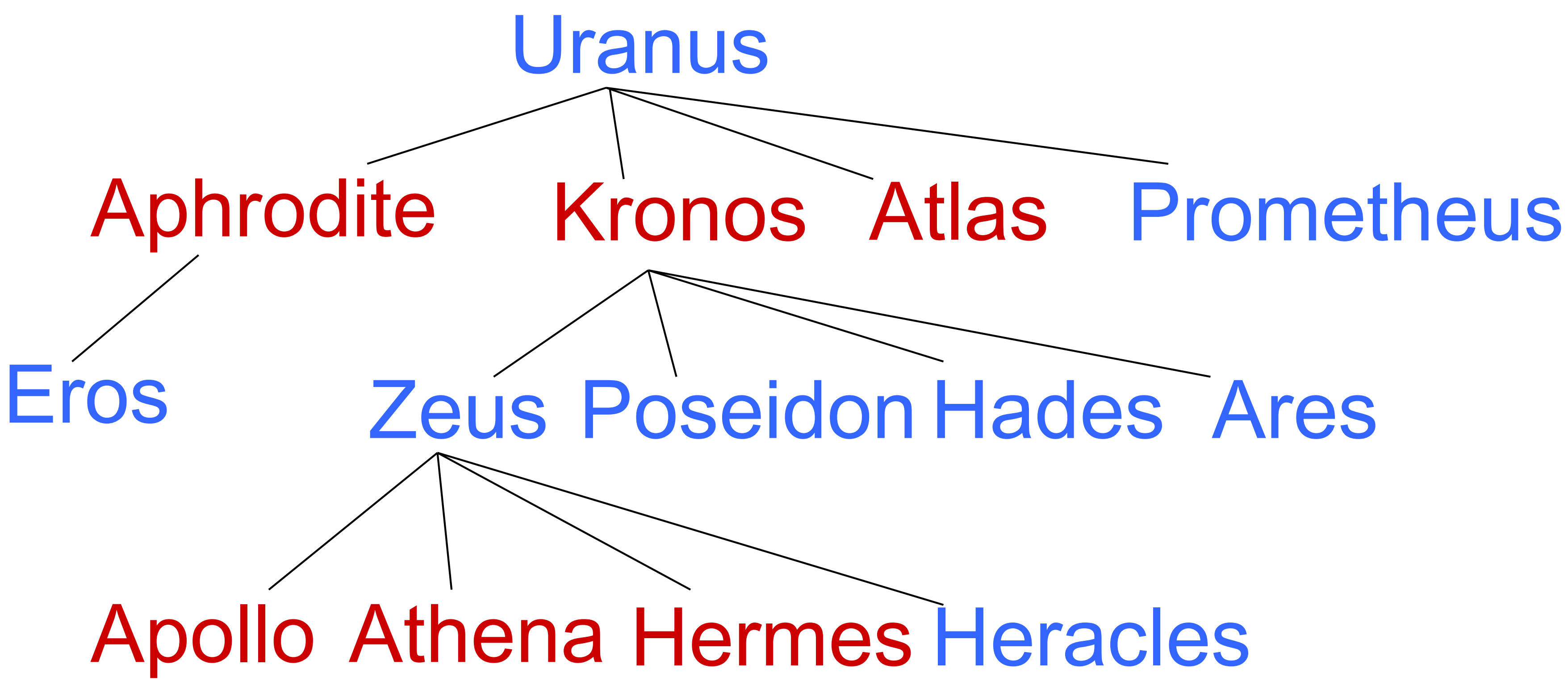
- A lógica proposicional fornece uma ferramenta formal para representar conhecimento e fazer inferências sobre ele.
- Porém, a lógica proposicional não consegue expressar fatos genéricos de modo conciso.

➔**Lógica de Primeira Ordem (FOL) ou**

➔**Lógica de Predicados**

LÓGICA DE PREDICADOS
(LÓGICA DE PRIMEIRA ORDEM – FOL)

Exemplo - Família dos Deuses Gregos



Lógica de Predicados

Permite representar:

- **Objetos**
  - exemplo: Aphrodite e Kronos
- **Relações** entre objetos
  - exemplo: Aphrodite é irmã de Kronos
  - Irmao(Aphrodite, Kronos)
- **Funções** de objetos
  - exemplo: Uranus é pai de Aphrodite
  - Pai(Aphrodite) = Uranus

Sintaxe do Cálculo de Predicados: **Símbolos**

- símbolos para **constantes**, que se referem a objetos, como Aphrodite e Zeus (maiúsculas);
- símbolos para **variáveis**, que se referem a objetos ou conjunto de objetos, como x, y, z ... (minúsculas);
- símbolos para **predicados**, que se referem a relações entre objetos, como Irmão, Primo;
- símbolos para **funções**, que se referem a funções de objetos, como Pai, Mãe.

Sintaxe do Cálculo de Predicados : **Termos**

Termos são expressões lógicas que se referem a um objeto.

- Podem ser constituídos por:
  - constantes, como Aphrodite e Zeus;
  - variáveis, como x, y, z, ….;
  - funções aplicadas a termos, como Pai(Aphrodite), Mãe(z), Mãe(Pai(Zeus)).

Sintaxe do Cálculo de Predicados em BNF

Símbolo inicial: <sentença>

<sentença>::= <sentença atômica> | ¬<sentença>|
(<sentença> <conectivo> <sentença>) |
<quantificador> <variável> <sentença>

<sentença atômica>::= <predicado> (<termo>, ...)

<termo> ::= <função> (<termo>, ...) | <constante> |
<variável>

<conectivo> ::= → | ∧ | ∨ | ↔

<quantificador> ::= ∀ | ∃

<constante> ::= A | X_1 | João | ...

<variável> ::= x | y | z | ...

<predicado> ::= Antes | TemCor | Chovendo | ...

<função> ::= Mãe | NUSP | ...

Quantificadores

Quantificador Universal

Sintaxe: $\forall x\ P(x)$

Lê-se para todo x, P(x) ou para qualquer x, P(x).

Quantificador Existencial

Sintaxe: $\exists x\ P(x)$

Lê-se para algum x, P(x) ou existe x, P(x).

Quantificadores – Exemplos de expressões

- Todo homem é mortal
  $\forall x\ (Homem(x) \rightarrow Mortal(x))$

- Nenhum homem é imortal
  $\forall x\ (Homem(x) \rightarrow \neg Imortal(x))$

- Pelo menos um homem é inteligente
  $\exists x\ (Homem(x) \wedge Inteligente\ (x))$

$\forall$ com $\rightarrow$ $\exists$ com $\wedge$

Quantificadores – Exemplos de expressões

- Todo homem ama as mulheres.
- Todo homem ama uma mulher.
- Há uma mulher amada por todos os homens.

Quantificadores – Exemplos de expressões

- Todo homem ama as mulheres.

  $\forall x\ \forall y\ (Homem(x) \land Mulher(y) \rightarrow Ama(x, y))$

- Todo homem ama uma mulher.

  $\forall x\ \exists y\ (Homem(x) \land Mulher(y) \rightarrow Ama(x, y))$

- Há uma mulher amada por todos os homens.

  $\exists y\ \forall x\ (Homem(x) \land Mulher(y) \rightarrow Ama(x, y))$

Quantificadores

Teorema:

Generalização da **Lei de De Morgan** para os quantificadores:

$$\neg \forall x\, P(x) \equiv \exists x \neg P(x)$$

$$\neg \exists x\, P(x) \equiv \forall x \neg P(x)$$

Um exemplo intuitivo seria:

- "Não é todo homem que é egoísta" equivale a "Existe pelo menos um homem que não é egoísta"

Semântica: Fórmulas Bem Formadas

- Definida do mesmo modo que no cálculo proposicional.
- fbfs podem ser **válidas**, **inválidas**, **contingentes**, **satisfazíveis** ou **insatisfazíveis**.

- A única diferença diz respeito à definição da semântica dos quantificadores.

Inferência

- Semelhante ao visto em lógica proposicional, porém mais complexo pela:
  - presença dos quantificadores
  - presença de variáveis

- Leva em conta duas noções fundamentais:
  - substituição
  - unificação

Inferência: **Substituição**

- Cada variável é associada no máximo com uma única expressão
- Nenhuma variável com uma expressão associada ocorre no escopo de qualquer outra expressão
  - ex: {x/A, y/F(B), z/w} é uma substituição
    {x/G(y), y/F(x)} não é uma substituição

- A notação utilizada é $\phi\sigma$, onde se aplica a substituição σ à expressão ϕ
  - ex: P(x, x, y, v) {x/A, y/F(B), z/w}
    resulta em:
    P(A, A, F(B), v)

Inferência: **Unificação**

Determina se duas expressões podem se tornar <u>idênticas</u> se suas variáveis forem substituídas de modo apropriado.

ex: $P(A, y, z)\ \{x/A, y/B, z/C\} =$

$P(A, B, C) =$

$P(x, B, z)\ \{x/A, y/B, z/C\}$

Exemplo: Resolução

Dado o domínio dos Deuses Gregos, considere as sentenças:

1. Todos os homens são mortais;
2. Kronos é um homem;

Provar que Kronos é mortal, utilizando refutação por resolução.

Exemplo: Resolução

1. $\forall x$ Homem (x) $\rightarrow$ Mortal (x)
2. Homem (Kronos)

Deseja-se provar Mortal (Kronos)

Passando para a forma clausal e eliminando $\forall$:

Exemplo: Resolução

1. $\forall x$ Homem (x) $\rightarrow$ Mortal (x)
2. Homem (Kronos)

Deseja-se provar Mortal (Kronos)

Passando para a forma clausal:

1. $\neg$Homem (x)$\vee$ Mortal (x)	Premissa
2. Homem (Kronos)	Premissa
3. $\neg$ Mortal(Kronos)	Neg. Conclusão
4. Mortal (Kronos)	RES 1, 2 {x/Kronos}
5. {}	RES 3, 4

Referências Bibliográficas

- S. Russel and P. Norvig. Inteligência Artificial. Tradução da 3a. Edição. 2013. Capítulos 7, 8 e 9.