# Agentes Lógicos

## Inteligência Artificial

## PCS3438

*Escola Politécnica da USP*

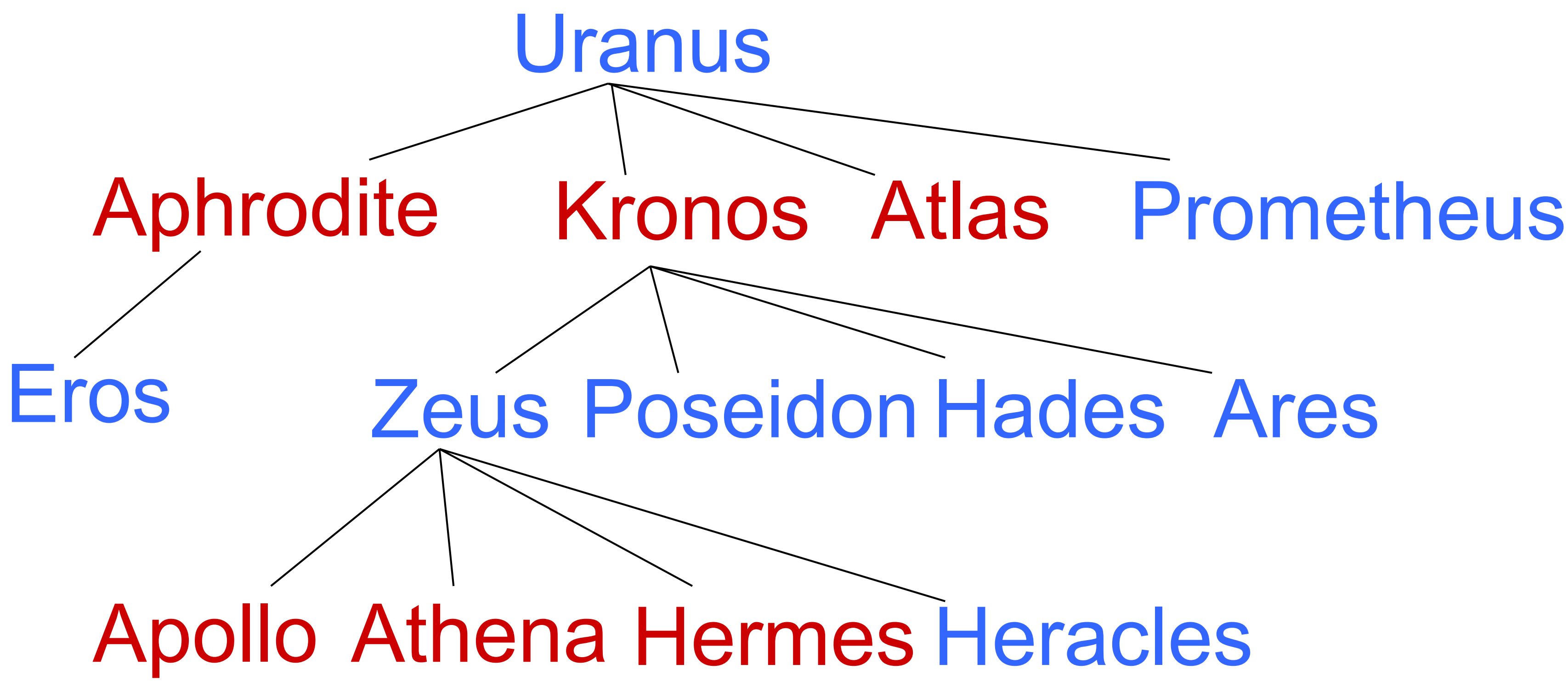
*Engenharia de Computação (PCS)*

# LÓGICA DE PRIMEIRA ORDEM

# Limites da lógica proposicional

- Apesar de ser simples, a lógica proposicional não consegue expressar fatos genéricos de modo conciso, como por exemplo:
  - “se o agente estiver em qualquer local com um leão à sua frente, não prosseguir em frente”.
- Seria interessante poder expressar tais fatos genéricos através de um linguagem que pudesse fazer referência a objetos e a algumas de suas propriedades.

# Exemplo - Família dos Deuses Gregos



# Lógica de Predicados

Permite representar:

- Objetos
  - exemplo: Aphrodite e Kronos
- Relações entre objetos
  - exemplo: Aphrodite é irmã de Kronos
- Funções de objetos
  - exemplo: Uranus é pai de Aphrodite

# Lógica de Predicados

Irmão (Kronos, Aphrodite) ∧
Irmão (Kronos, Atlas)

Irmão (Atlas, Aphrodite)

Consequência Lógica

Semântica

Representação

Semântica

Mundo

Decorrem

# Comparação entre Linguagens Lógicas

- Linguagens lógicas têm compromissos:
  - **Ontológicos**: como representar o mundo
  - **Epistemológicos**: como um agente crê nestes fatos

| Linguagem | Comp. Ontológico | Comp. Epistem. |
|---|---|---|
| L. Proposicional | fatos | V, F, desconhecido |
| L. Predicados | fatos, objetos, relações | V, F, desconhecido |
| L. Temporal | fatos, objetos, relações, tempo | V, F, desconhecido |
| Teoria Probabilidades | fatos | graus de crença $\in$ [0,1] |
| L. Nebulosa | graus de verdade $\in$ [0,1] | graus de crença $\in$ [0,1] |

# Sintaxe do Cálculo de Predicados: Símbolos

Os elementos básicos da sintaxe do cálculo de predicados são **<u>símbolos</u>** que se referem a objetos, relações e funções:

- 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.

Termos 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: Predicados

Uma sentença atômica será composta por um predicado,  aplicado a um ou mais termos, dependendo de sua aridade:

- Irmão (Kronos, Aphrodite)
- Irmão (Pai (Zeus), Mãe (Eros))
- Irmão (x, y)

Fórmulas bem formadas serão compostas por predicados ligados através de conectivos lógicos:

- Irmão(Pai(Zeus), Mãe(Eros))→Primo(Zeus, Eros)

# 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> ::= → | ∧ | v | ↔

<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
  $\exists x\ (Homem(x) \wedge Inteligente\ (x))$

# 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"

# Quantificadores: Variáveis Livres e Ligadas

No cálculo de predicados, devido à presença dos quantificadores, uma variável pode estar:

- **livre** (*free*)
  - Irmão(x,y) → ¬FilhoÚnico(x)

- **ligada** (*bounded*)
  - ∀x ∀y (Irmão(x,y) → ¬FilhoÚnico(x))

Estando ligada, diz-se que uma variável está no **escopo** de um quantificador.

# Quantificadores: Sentenças Abertas, Fechadas e Primitivas

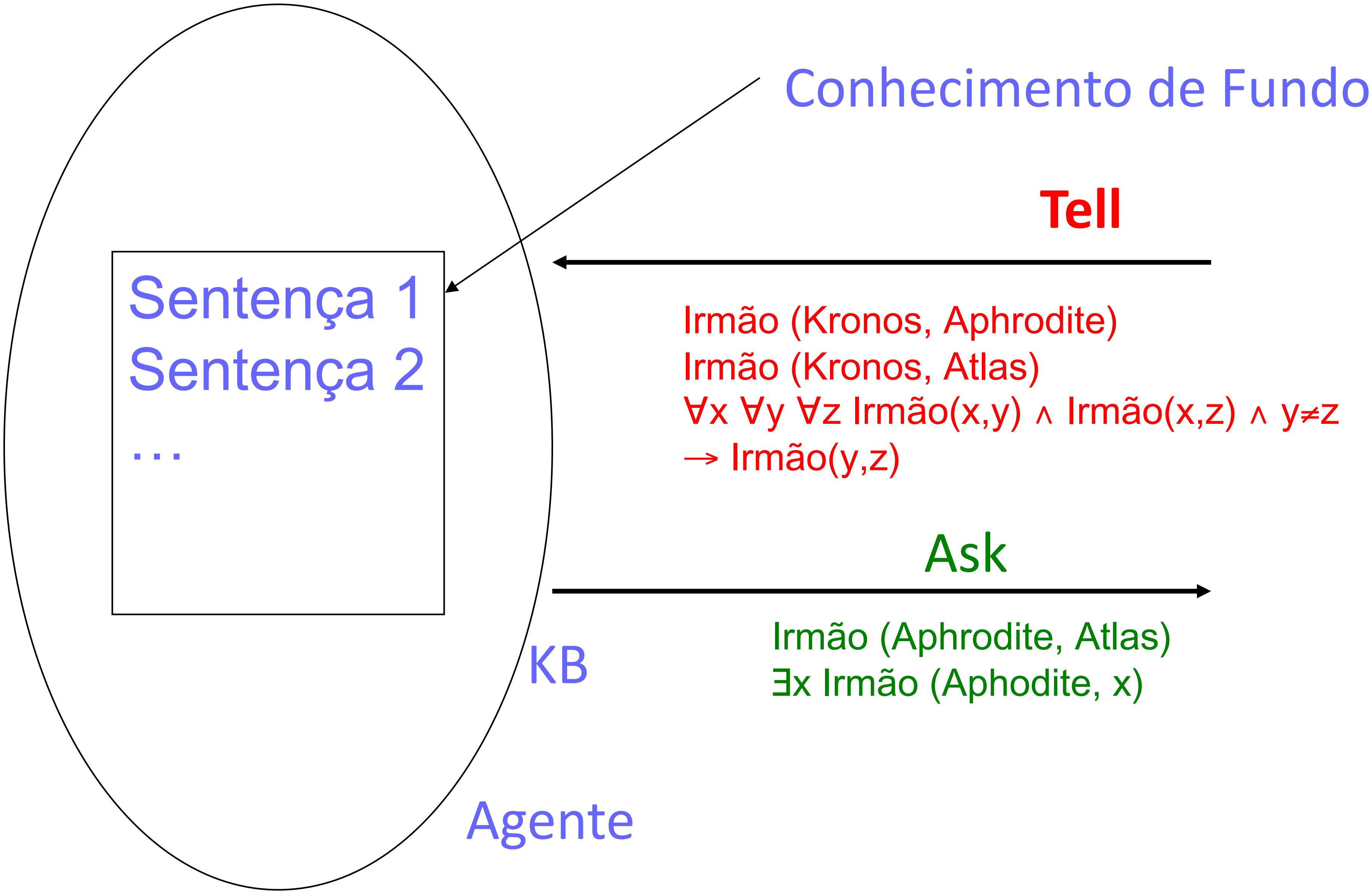
No cálculo de predicados, uma sentença pode ser:

- **aberta**: sem restrições
  - Irmão (x, y) → MesmoPai (x, y)
- **fechada** (*closed*): sem variáveis livres
  - ∃x Irmão (Kronos, x)
- **primitiva** (*grounded*): sem variáveis livres ou ligadas
  - Irmão(Kronos,Aphrodite) → MesmoPai(Kronos, Aphrodite)

# Semântica: Fórmulas Bem Formadas

- A semântica das fórmulas bem formadas, construídas utilizando os conectivos lógicos, é definida do mesmo modo que no cálculo proposicional.
- De modo semelhante, fbfs podem ser **válidas**, **inválidas**, **contingentes**, **satisfazíveis** ou **insatisfazíveis**.
- A única diferença diz respeito à definição da semântica dos quantificadores.

# Agente Baseado em Lógica de Predicados



# Inferência

- O procedimento de inferência em lógica de predicados é semelhante ao visto em lógica proposicional, porém mais complexo pela:
  - presença dos quantificadores
  - presença de variáveis
- Este procedimento leva em conta duas noções fundamentais:
  - substituição
  - unificação

# Inferência: Substituição

Uma **substituição** é um conjunto finito de associações entre variáveis e expressões tais que:

- 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

# Inferência: Substituição

Uma substituição pode ser aplicada em uma fbf do cálculo de predicados, gerando uma instância desta substituição.

A notação utilizada é $\phi\sigma$, onde se aplica a substituição $\sigma$ à expressão $\phi$

– ex: P(x, x, y, v) {x/A, y/F(B), z/w}

resulta em:

P(A, A, F(B), v)

# Inferência: Unificação

Trata-se do processo que determina se duas expressões podem se tornar idênticas se suas variáveis forem substituídas de modo apropriado.

Um conjunto de expressões $\{\phi_1, \phi_2, ..., \phi_n\}$ é **unificável** se e somente se existir uma substituição $\sigma$ que as torna idênticas:

$\phi_1\sigma = \phi_2\,\sigma = ... = \phi_n\,\sigma$. Neste caso, $\sigma$ é dito um **unificador** do conjunto.

– 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}

# Regras de Inferência para Cálculo de Predicados

- Instanciação Universal $\forall x\ \alpha \models \alpha\ \{x/g\}$
  - O termo g deve ser um termo primitivo
- Instanciação Existencial $\exists x\ \alpha \models \alpha\ \{x/c\}$
  - A constante c não deve ter aparecido na KB

Com estas regras, pode-se construir uma KB proposicional, já que todas as sentenças serão primitivas!

# Regras de Inferência para Cálculo de Predicados – Exemplo

$\forall x\ \forall y\ \forall z\ Pai(x,y) \rightarrow Filho(y,x)$

Pai(Ur, Ap)
Pai(Ur, Kr)

{x/Ur,y/Ur}

{x/Ur,y/Ap}

{x/Ur, y/Kr}

Pai(Ur,Ur) → Filho (Ur,Ur)

Pai(Ur,Ap) → Filho (Ap,Ur)

Pai(Ur,Kr) → Filho (Kr,Ur)

Pai(Ur, Ap)

Pai(Ur, Kr)

# Regras de Inferência para Cálculo de Predicados

## Modus Ponens Generalizado

$$p1', p2', \ldots, pn' \quad p1 \wedge p2 \wedge \ldots \wedge pn \rightarrow q \models q\sigma,$$

*onde* $pi\,\sigma = pi'\,\sigma,\ i \in \{1..n\}$

- Mais eficiente, pois não preciso gerar sentenças desnecessárias
- Existem algoritmos eficientes para achar os unificadores. Em particular, existem infinitos unificadores para duas sentenças, e sempre busca-se o unificador mais geral (mgu).

# Exemplo : Modus Ponens Generalizado

Dado o domínio dos Deuses Gregos, considere as sentenças:

1. Uma pessoa é avô de outra se for pai de seu pai ou de sua mãe;
2. Kronos é pai de Zeus;
3. Uranus é pai de Kronos.

Provar que Uranus é avô de Zeus.

# Exemplo : Modus Ponens Generalizado

1. $\forall x \forall y \forall z\ Pai(x,y) \land (Pai(y,z) \lor Mãe(y,z)) \rightarrow Avô(x, z)$ premissa
2. Pai (Kronos, Zeus) premissa
3. Pai (Uranus, Kronos) premissa
4. Pai (Kronos, Zeus) ∨ Mãe(Kronos, Zeus) AD 2
5. Pai(Uranus,Kronos) ∧ (Pai(Kronos,Zeus)∨Mãe(Kronos,Zeus)) CJ 3, 4
6. Avô (Uranus, Zeus) MPG 1,5 {x/Uranus, y/Kronos, z/Zeus}

# RESOLUÇÃO PARA LÓGICA DE PREDICADOS

# Transformação para Forma Clausal

1. Substituir $\alpha \leftrightarrow \beta$ por $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$ e substituir $\alpha \rightarrow \beta$ por $\neg\alpha \vee \beta$
2. Colar as negações nos átomos, utilizando as equivalências $\neg(\neg\alpha) \equiv \alpha$, $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$, $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$, $\neg\exists x\, \alpha \equiv \forall x\, \neg\alpha$, e $\neg \forall x\, \alpha \equiv \exists x\, \neg\alpha$.
3. Padronizar as variáveis, trocando os nomes quando estas aparecem no escopo de quantificadores diferentes
4. Remover os quantificadores existenciais utilizando variáveis e funções de Skolem
5. Remover os quantificadores universais
6. Distribuir as disjunções pelas conjunções, utilizando $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

# Transformação para Forma Clausal

Prova-se que qualquer fbf do cálculo de predicados pode ser transformada num conjunto de cláusulas equivalente, através de uma sequência definida de passos.

Exemplo: "Todo aquele que ama todos os animais é amado por alguém"

Como seria a representação disto em lógica de predicados?

$$\forall x\ (\forall y\ (Animal(y) \rightarrow Loves(x,y))) \rightarrow (\exists y\ (Loves(y, x)))$$

# Transformação para Forma Clausal

$\forall x\ (\forall y\ Animal(y) \rightarrow Loves(x,y)) \rightarrow (\exists y\ Loves(y, x))$

1. Substituir $\alpha \leftrightarrow \beta$ por $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$ e substituir $\alpha \rightarrow \beta$ por $\neg\alpha \vee \beta$

$\forall x\ \neg\ (\forall y\ \neg Animal(y) \vee Loves(x,y)) \vee (\exists y\ Loves(y, x))$

2. Colar as negações nos átomos, utilizando as equivalências $\neg(\neg\alpha) \equiv \alpha$, $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$, $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$, $\neg\exists x\ \alpha \equiv \forall x\ \neg\alpha$, e $\neg\forall x\ \alpha \equiv \exists x\ \neg\alpha$.

$\forall x(\exists\ y \neg\ (\neg Animal(y) \vee Loves(x,y))) \vee (\exists y\ Loves(y, x))$

$\forall x(\exists\ y\ \neg\neg Animal(y) \wedge \neg Loves(x,y)) \vee (\exists y\ Loves(y, x))$

$\forall x(\exists\ y\ Animal(y) \wedge \neg Loves(x,y)) \vee (\exists y\ Loves(y, x))$

# Transformação para Forma Clausal

$$\forall x(\exists y\, Animal(y) \land \neg Loves(x,y)) \lor (\exists y\, Loves(y, x))$$

3. Padronizar as variáveis, trocando os nomes quando estas aparecem no escopo de quantificadores diferentes

$$\forall x(\exists y\, Animal(y) \land \neg Loves(x,y)) \lor (\exists z\, Loves(z, x))$$

4. Remover os quantificadores existenciais utilizando variáveis e funções de Skolem

$\forall x(Animal(A) \land \neg Loves(x,A)) \lor Loves(B, x)$ ***ERRO!!!***

$$\forall x(Animal(F(x)) \land \neg Loves(x,F(x))) \lor Loves(G(x), x)$$

5. Remover os quantificadores universais

$$(Animal(F(x)) \land \neg Loves(x,F(x))) \lor Loves(G(x), x)$$

# Transformação para Forma Clausal

$\boxed{(Animal(F(x)) \wedge \neg Loves(x,F(x))) \vee Loves(G(x), x)}$

6. Distribuir as disjunções pelas conjunções, utilizando $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

$(Animal(F(x)) \vee Loves(G(x), x)) \wedge$

$(\neg Loves(x,F(x)) \vee Loves(G(x), x))$

Esta sentença gerou 2 cláusulas na KB

# Mais um exemplo: função de Skolem

- Quando um quantificador existencial estiver aninhado internamente a um quantificador universal, devemos usar funções de Skolem (caso contrário, basta substituir $\exists x\ P(x)$ por $P(A)$, sendo A uma constante que ainda não apareceu em lugar algum da KB):

$$\forall x\ Person(x) \rightarrow \exists y\ Heart(y) \wedge Has(x,y)$$

- **ERRO**: $\forall x\ Person(x) \rightarrow Heart(H) \wedge Has(x,H)$ pois indica que <u>todos</u> têm o <u>mesmo</u> coração H!
- **CORRETO**: $\forall x\ Person(x) \rightarrow Heart(H(x)) \wedge Has(x,H(x))$

# Regras de Inferência para Cálculo de Predicados

Resolução

$$p1 \vee \ldots \vee pi \vee \ldots \vee pn, \ \ m1 \vee \ldots \vee mi \vee \ldots mn$$

$$\models$$

$$(p1 \vee \ldots \vee pi\text{-}1 \vee pi\text{+}1 \vee \ldots \vee pn \vee$$
$$m1 \vee \ldots \vee mi\text{-}1 \vee mi\text{+}1 \vee \ldots \vee mn)\ \sigma$$

onde $pi\ \sigma = \neg mi\ \sigma$

# Regras de Inferência para Cálculo de Predicados

Exemplo:

$Animal(F(x)) \vee Loves(G(x), x)$ $\quad$ $\neg Loves(u, v) \vee \neg Kills(u, v)$

unificam com a substituição $\{u/G(x), v/x\}$ e geram o resolvente:

$$Animal(F(x)) \vee \neg Kills(G(x), x)$$

# 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:

| | |
|---|---|
| 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 |

# Inferência: Complexidade

O problema geral de decidir se uma base de conhecimento KB em lógica de predicados implica logicamente uma fórmula $\alpha$ **não é decidível**:

- para **alguns** conjuntos de sentenças KB, garante-se que o procedimento de prova encontra uma prova de $\alpha$ ou de $\neg\alpha$
  - problema da implicação lógica é **decidível**
- em outros casos, quando nem $\alpha$ nem $\neg\ \alpha$ são logicamente implicados por KB, o procedimento nunca termina!

# Referências Bibliográficas


- S. Russel and P. Norvig.
  Artificial Intelligence: A Modern Approach.
  Chapter 8 and 9.