Lógica de predicados - Inferências

- Regras de inferência permitem concluir a veracidade de sentenças a partir da veracidade de sentenças conhecidas
  - Por exemplo, podemos inferir que Perveso(João) é verdade
  - A partir das sentenças:

$$\forall x \; Rei(x) \wedge Ambicioso(x) \rightarrow Perverso(x)$$
$$Rei(João)$$
$$Ambicioso(João)$$

  - Fazendo a substituição {x/João}

  - A regra de inferência que permite esse processo é a regra chamada Modus Ponens Generalizado

Unificação e Substituição

- Substituição:
  - Conjunto de pares variável-termo
- Unificação:
  - Processo que encontra substituições que façam expressões lógicas diferentes parecerem idênticas
- Algoritmo de unificação:
  - Recebe duas sentenças e retorna um unificador para elas, se existir algum
- UNIFICAR(p,q) = θ onde SUBST(θ,p) = SUBST(θ,q)

Algoritmo de Unificação – Exemplos

- Conhece(João,x) - Conhece(João,Jane)
- Unificador: {x/Jane}

- Conhece(João,x) - Conhece(y, Bill)
- Unificador: {x/Bill, y/João}

- Conhece(João,x) - Conhece(y, Mãe(y))
- Unificador: {y/João, x/Mãe(João)}

- Conhece(João,x) - Conhece(x, Elizabeth)
- Falha
- O problema ocorre porque as variáveis tem o mesmo nome. Pode ser evitado renomeando as variáveis (padronização separada)

Lógica de predicados - Inferências

- Modus Ponens Generalizado:

  - Para as sentenças atômicas p_i, p'_i e q, em que exista uma substituição θ tal que

$$SUBST(\theta, p_i) = SUBST(\theta, p'_i) \text{ para todo } i,$$

$$\frac{p1', p2', \ldots, pn' \quad (p1 \wedge p2 \wedge \ldots \wedge pn \rightarrow q)}{SUBST(\theta, q)}$$

Lógica de predicados - Inferências

- Modus Ponens Generalizado:

$$\frac{p1',p2',(p1 \wedge p2 \rightarrow q)}{SUBST(\theta,q)}$$

A partir da cláusulas
$\forall$x Rei(x) $\wedge$ Ambicioso(x) $\rightarrow$ Perverso (x)
Rei(João)
Ambicioso(João)

A regra de inferência Modus Ponens generalizado permite inferir:
Perverso (João)

p1´: Rei(João) p1: Rei(x)
p2´: Ambicioso(João) p2: Ambicioso(x)
θ : {x/João} q: Perverso (x)
SUBST(θ,q): Perverso(João)

Lógica de predicados - Inferências

- Modus Ponens Generalizado:

$$\frac{p1', p2', (p1 \wedge p2 \rightarrow q)}{SUBST(\theta, q)}$$

Se as cláusulas fossem:

$\forall$x Rei(x) $\wedge$ Ambicioso(x) $\rightarrow$ Perverso (x)

Rei(João)

$\forall$y Ambicioso(y)

A regra de inferência Modus Ponens generalizado permite inferir:

Perverso (João)

p1´: Rei(João) **p1: Rei(x)**

p2´: Ambicioso(y) **p2: Ambicioso(x)**

θ : {x/João, y/João} **q: Perverso (x)**

SUBST(θ,q): Perverso(João)

Algoritmos de Inferência na Lógica de Predicados

- A lógica de primeira ordem é um formalismo poderoso para representar conhecimento e raciocinar sobre ele, por meio das inferências

- As inferências, porém, devem ser aplicadas por meio de mecanismos sistemáticos.

- Existem três famílias de algoritmos de inferência de primeira ordem:
  - Encadeamento para frente
  - Encadeamento para trás
  - Regra da resolução

Algoritmos de Inferência na Lógica de Predicados

- **Encadeamento para frente** é uma forma de **raciocínio dirigido por dados** – que começa com dados conhecidos e caminha para frente até provar um objetivo

- **Encadeamento para trás** é uma forma de **raciocínio dirigido por objetivos** – começa com o objetivo que se deseja provar e caminha para trás, até encontrar fatos que dêem suporte ao objetivo

- Os algoritmos de encadeamento para frente e para trás são aplicáveis a uma forma restrita de cláusulas da lógica de primeira ordem: as **cláusulas definidas de primeira ordem**

Cláusulas Definidas de Primeira Ordem

- São disjunções de literais dos quais exatamente um é positivo

  $\neg$ Rei(x) $\vee$ $\neg$ Ambicioso(x) $\vee$ Perverso(x)

  - Literal: sentença atômica ou sua negação

- Assim, uma CDPO é atômica ou é uma implicação cujo antecedente é uma conjunção de literais positivos e cujo consequente é um único literal positivo.

- Pelas regras de equivalência da lógica $(\neg P \vee Q) \equiv (P \rightarrow Q)$

- **($\neg$ Rei(x) $\vee$ $\neg$ Ambicioso(x) $\vee$ Perverso(x)) $\equiv$ (Rei(x) $\wedge$ Ambicioso(x) $\rightarrow$ Perverso (x))**

- As variáveis dos literais são quantificadas universalmente mas o símbolo é omitido

  Rei(x) $\wedge$ Ambicioso(x) $\rightarrow$ Perverso (x)
  Rei(João)
  Ambicioso(João)

Representação do Conhecimento e Inferência – Exemplo

Extraído de Russell& Norvig, Inteligência Artificial, 2ª. Ed., capítulo 9, seções 9.2 e 9.3

- Dado um problema, construir a base de conhecimento para representar esse problema e responder consultas referentes a ele

- Problema:
  - A lei diz que é crime um americano vender armas a nações hostis. O país Nono, inimigo da América, tem alguns mísseis, e todos foram vendidos pelo Coronel West, um americano.

- Consulta:
  - West é criminoso?

Definição de cláusulas para o problema

- O primeiro passo é representar os fatos como cláusulas definidas de primeira ordem

"É crime um americano vender armas a nações hostis"

$Americano(x) \land Arma(y) \land Vende(x,y,z) \land Hostil(z) \rightarrow Criminoso(x)$

"Nono tem alguns mísseis"

$\exists x\ Possui(Nono,x) \land Míssil(x)$

- CDPO não podem conter quantificador existencial, que deve ser eliminado por uma regra chamada instanciação existencial
- A variável é substituída por um novo símbolo de constante único
- O novo símbolo é chamado de constante de Skolem

Definição de cláusulas para o problema

- O primeiro passo é representar os fatos como cláusulas definidas de primeira ordem

“Nono tem alguns mísseis”

A sentença

$\exists$x Possui(Nono,x) $\wedge$ Míssil(x)

é transformada em:

Possui(Nono, M1)

Míssil(M1)   (M1 é uma constante que não aparece na BC)

Definição de cláusulas para o problema

“Todos foram vendidos pelo Coronel West”

Míssil(x) $\wedge$ Possui(Nono, x) $\rightarrow$ Vende(West, x, Nono)

“Mísseis são armas”

Míssil(x) $\rightarrow$ Arma(x)

Um inimigo da América é considerado hostil

Inimigo(x,América) $\rightarrow$ Hostil(x)

“West, um americano...”

Americano(West)

“O país Nono, inimigo da América”

Inimigo(Nono,América)

Base de Conhecimento

1. Americano(x) $\wedge$ Arma(y) $\wedge$ Vende(x,y,z) $\wedge$ Hostil(z) $\rightarrow$ Criminoso(x)
2. Possui(Nono, M1)
3. Míssil(M1)
4. Míssil(x) $\wedge$ Possui(Nono, x) $\rightarrow$ Vende(West, x, Nono)
5. Míssil(x) $\rightarrow$ Arma(x)
6. Inimigo(x,América) $\rightarrow$ Hostil(x)
7. Americano(West)
8. Inimigo(Nono,América)

Esta base de conhecimento não tem nenhum símbolo de função (base de conhecimento Datalog). A inferência é mais simples sem símbolos de função.