

# Inteligência Artificial

## IA

Prof. João Luís Garcia Rosa

## IV. RACIOCÍNIO BASEADO EM REGRAS

Parte 1  
2004

### Introdução

- A forma como um corpo de conhecimento sobre um certo campo é expresso por um especialista deste campo, freqüentemente contém informação importante sobre como esse conhecimento pode ser usado da melhor forma. Suponha, por exemplo, que um matemático diga:  
“Se  $X$  e  $Y$  são ambos maiores que zero, o produto de  $X$  e  $Y$  também é maior que zero.”
- No cálculo de predicados esta sentença ficaria:  
$$\forall X \forall Y ((m(X,0) \wedge m(Y,0)) \rightarrow m(\text{vezes}(X, Y), 0)).$$
- Entretanto, pode-se usar a seguinte fórmula equivalente:  
$$\forall X \forall Y ((m(X,0) \wedge \neg m(\text{vezes}(X, Y), 0)) \rightarrow \neg m(Y,0)).$$

## Introdução

- O conteúdo lógico da sentença do matemático é independente das muitas formas equivalentes do cálculo de predicados que poderiam representá-la. Mas, na forma que as sentenças do Português são construídas, freqüentemente contêm informação de controle extra-lógica ou heurística. No exemplo acima, a sentença parece indicar que se está habituado ao fato de que se  $X$  e  $Y$  são individualmente maiores que zero, então é óbvio provar que  $X$  multiplicado por  $Y$  é maior que zero.

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 3

## Introdução

- Muito do conhecimento usado por sistemas de IA é diretamente representável por expressões gerais de implicação. Veja as seguintes sentenças:
  - (1) Todos os vertebrados são animais.  
 $\forall X (\text{vertebrado}(X) \rightarrow \text{animal}(X))$
  - (2) Todos do departamento de computação acima de 30 anos são casados.  
 $\forall X \forall Y ((\text{trabalha\_em}(\text{dep\_computação}, X) \wedge \text{idade}(X, Y) \wedge \text{ma}(Y, 30)) \rightarrow \text{casado}(X))$
  - (3) Existe um cubo acima de todo cilindro vermelho.  
 $\forall X ((\text{cilindro}(X) \wedge \text{vermelho}(X)) \rightarrow \exists Y (\text{cubo}(Y) \wedge \text{acima}(Y, X)))$

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 4

## Introdução

- O sistema descrito aqui não converte fórmulas em cláusulas; eles as usam numa forma perto da sua forma original dada. As fórmulas que representam conhecimento de asserção sobre o problema são separadas em duas categorias: *regras* e *fatos*. As regras consistem das asserções dadas na forma implicacional. Tipicamente, elas expressam conhecimento *geral* sobre uma área particular e são usadas como regras de produção. Os fatos são as asserções que não são expressas como implicações. Tipicamente, eles representam conhecimento *específico* relevante a um caso particular. A tarefa dos sistemas de produção é provar uma *fórmula meta* a partir destes fatos e regras.

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 5

## Introdução

- Em sistemas por encadeamento *progressivo* (*forward* ou “data-driven”), as implicações usadas como regras-P operam numa base de dados global de fatos até que uma condição de terminação envolvendo a fórmula meta seja alcançada. Em sistemas por encadeamento *regressivo* (*backward* ou “goal-driven”) as implicações usadas como regras-R operam numa base de dados global de metas até que uma condição de terminação envolvendo os fatos seja alcançada. Combinar operação progressiva e regressiva também é possível.

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 6

## Introdução

- Este tipo de sistema de prova de teorema é um sistema *direto* ao contrário do sistema de refutação. Um sistema direto não é necessariamente mais eficiente que um sistema de refutação, mas sua operação parece ser intuitivamente mais fácil para as pessoas entenderem.
- Sistemas deste tipo são freqüentemente chamados de *sistemas de dedução baseados em regras*, para enfatizar a importância de se usar regras para fazer deduções. A pesquisa em IA tem produzido muitas aplicações de sistemas baseados em regras.

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 7

## Um Sistema de Dedução Progressivo

- *A forma E/OU para Expressões de Fatos*
  - O sistema progressivo tem como sua base de dados global inicial uma representação para o conjunto de fatos dado. Em particular, não se pretende converter estes fatos em forma de cláusulas. Os fatos são representados como fórmulas do cálculo de predicados que foram transformadas em formas livres de implicações chamadas *formas E/OU*. Para converter uma fórmula na forma E/OU, os símbolos “ $\rightarrow$ ” (se existirem) são eliminados, usando a equivalência  $W1 \rightarrow W2 \equiv \neg W1 \vee W2$ . (Tipicamente, existem poucos símbolos “ $\rightarrow$ ” entre os fatos porque as implicações são preferivelmente representadas como regras.) Depois, os símbolos de negação são movidos para dentro (usando as leis de De Morgan) até que seus escopos incluam, no máximo, um único predicado. A expressão resultante é então Skolemizada e prenexada; as variáveis dentro dos escopos dos quantificadores universais são padronizadas através da renomeação, as variáveis quantificadas existencialmente são substituídas por funções de Skolem, e os quantificadores universais são eliminados. Qualquer variável restante é assumida ter quantificação universal. Ou seja, na verdade é aplicado o algoritmo da representação clausal até o passo imediatamente anterior a obtenção da forma normal conjuntiva, complementando com a eliminação dos quantificadores universais.

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 8

## Um Sistema de Dedução Progressivo

- Por exemplo, a expressão fato:

$$\exists U \forall V (q(V, U) \wedge \neg ((r(V) \vee p(V)) \wedge s(U, V)))$$

- é convertida para

$$q(V, a) \wedge ((\neg r(V) \wedge \neg p(V)) \vee \neg s(a, V))$$

- As variáveis podem ser renomeadas de tal forma que a mesma variável não ocorra em conjunções diferentes (principais) da expressão fato. A renomeação de variáveis neste exemplo leva à expressão:

$$q(W, a) \wedge ((\neg r(V) \wedge \neg p(V)) \vee \neg s(a, V)).$$

- Uma expressão na forma E/OU consiste de subexpressões de literais conectados por símbolos “ $\wedge$ ” e “ $\vee$ ”. Note que uma expressão na forma E/OU não está na forma de cláusula. Está muito mais perto da expressão original.

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 9

## Usando Grafos E/OU para Representar Expressões de Fatos

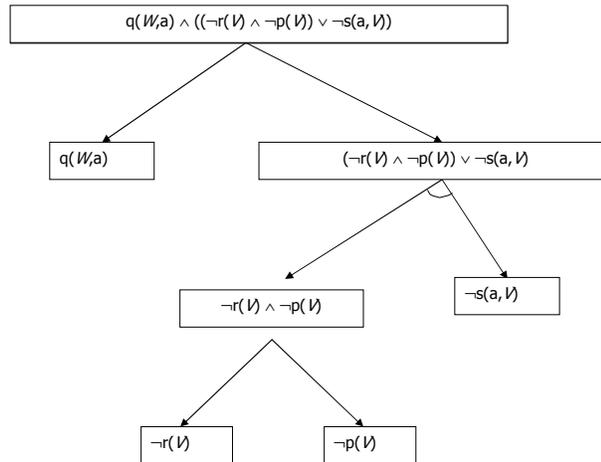
- Um grafo E/OU pode ser usado para representar uma expressão fato na forma E/OU. Por exemplo, a árvore E/OU a seguir representa a expressão fato posta na forma E/OU acima. Cada subexpressão da expressão fato é representada por um nó no grafo. As subexpressões relacionadas disjuntivamente,  $E_1, \dots, E_k$ , de um fato,  $(E_1 \vee \dots \vee E_k)$ , são representadas por nós descendentes conectados a seus nós pais por um conector-k (que contém um arco de ligação entre os descendentes). Cada subexpressão conjuntiva,  $E_1, \dots, E_n$ , de uma expressão,  $(E_1 \wedge \dots \wedge E_n)$ , é representada por um único nó descendente conectado ao nó pai por um conector-l.

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 10

## Usando Grafos E/OU para Representar Expressões de Fatos



João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 11

## Usando Grafos E/OU para Representar Expressões de Fatos

- Os nós folhas da representação de grafo E/OU de uma expressão fato estão rotulados pelos literais que ocorrem na expressão. Chama-se o nó no grafo que rotula a expressão fato inteira de *nó raiz*. O nó raiz não tem nenhum ancestral no grafo.
- Uma propriedade interessante da representação de grafo E/OU de uma fórmula é que o conjunto de cláusulas no qual a fórmula pode ser convertido, pode ser visto como o conjunto de grafos solução (terminando em nós folhas) do grafo E/OU. Então, as cláusulas que resultam da expressão  $q(W,a) \wedge ((\neg r(V) \wedge \neg p(V)) \vee \neg s(a, V))$  são:
  - $q(W,a)$
  - $\neg s(a, V) \wedge \neg r(V)$
  - $\neg s(a, V) \wedge \neg p(V)$

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 12

## Usando Regras para Transformar Grafos E/OU

- As regras de produção usadas pelo sistema de produção progressivo são aplicadas a estruturas de grafos E/OU para produzir estruturas de grafos transformadas. Estas regras são baseadas em fórmulas implicacionais que representam conhecimento assertional geral sobre um domínio de problema. Para simplificar, limitou-se os tipos de fórmulas permitidas como regras àquelas da forma:

$$L \rightarrow W,$$

- onde L é um literal único, W é uma fórmula arbitrária (assumida estar na forma E/OU), e quaisquer variáveis ocorrendo na implicação são assumidas ter quantificação universal sobre a implicação inteira. As variáveis nos fatos e regras são padronizadas de tal forma que nenhuma variável ocorra em mais que uma regra e que as variáveis das regras sejam diferentes das dos fatos.

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 13

## Usando a Fórmula Meta para Terminação

- O objetivo do sistema de produção progressivo descrito é provar alguma fórmula meta a partir de uma fórmula fato e de um conjunto de regras. Este sistema progressivo é limitado em relação ao tipo de expressões metas que ele pode provar; especificamente, ele pode provar apenas as fórmulas metas cuja forma seja uma *disjunção* de literais. Representa-se esta fórmula meta por um conjunto de literais e assume-se que os membros deste conjunto estão relacionados disjuntivamente. Os literais metas (assim como as regras) podem ser usados para adicionar descendentes ao grafo E/OU. Quando um dos literais metas casa com um literal rotulando um nó literal, n, do grafo, adiciona-se um novo descendente do nó n, rotulado pelo literal meta casado, ao grafo. Este descendente é chamado de *nó meta*. Os nós metas são conectados aos seus pais por "matching arcs". O sistema de produção termina de forma bem sucedida quando ele produz um grafo E/OU contendo um grafo de solução que termina em nós metas. (Na terminação, o sistema inferiu uma cláusula idêntica a alguma subparte da cláusula meta.)

João Luís G. Rosa

<http://docentes.puc-campinas.edu.br/ceatec/joaoluis/ia.html>

IA-2004-IV slide 14



## Receita para resolução por encadeamento progressivo

- Para resolver um problema através do encadeamento progressivo, é necessária a execução de alguns passos:
  - Verificar se os elementos estão no formato adequado: o fato deve estar na forma E/OU; as regras devem ter apenas um literal como antecedente e a meta deve ser uma disjunção de literais. Se a meta não obedecer a esta exigência, não é possível resolver através do encadeamento progressivo;
  - Construir o grafo E/OU para a expressão fato;
  - Construir o grafo do conhecimento para a expressão fato, aplicando as regras no grafo E/OU;
  - Obter as cláusulas-P a partir do grafo do conhecimento;
  - Verificar se a meta foi alcançada a partir do conjunto de cláusulas obtidas. Lembre-se de que este conjunto é uma conjunção de cláusulas-P, ou seja, uma conjunção de disjunções.