PARTE 5. SISTEMAS DE DEDUÇÃO BASEADOS EM REGRAS

5.1. Introdução

A forma como um pedaço 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:

$\forall x \forall y\ ((\mathrm{M}(x,0) \wedge \mathrm{M}(y,0)) \rightarrow \mathrm{M}(\mathrm{vezes}(x,y),0))$.

Entretanto, pode-se usar a seguinte fórmula equivalente:

$\forall x \forall y\ ((\mathrm{M}(x,0) \wedge \neg\mathrm{M}(\mathrm{vezes}(x,y),0)) \rightarrow \neg\mathrm{M}(y,0))$.

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.

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.
$\forall x\ (\mathrm{VERTEBRADO}(x) \rightarrow \mathrm{ANIMAL}(x))$

(2) Todos do departamento de computação acima de 30 anos são casados.
$\forall x \forall y\ ((\mathrm{TRABALHA\text{-}EM}(\mathrm{DEP\_COMPUTACAO},x) \wedge \mathrm{IDADE}(x,y) \wedge \mathrm{M}(y,30)) \rightarrow \mathrm{CASADO}(x))$

(3) Existe um cubo acima de todo cilindro vermelho.
$\forall x\ ((\mathrm{CILINDRO}(x) \wedge \mathrm{VERMELHO}(x)) \rightarrow \exists y\ (\mathrm{CUBO}(y) \wedge \mathrm{ACIMA}(y,x)))$

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.

Em sistemas *para frente* ("forward"), as implicações usadas como regras-F operam numa base de dados global de fatos até que uma condição de terminação envolvendo a fórmula meta seja alcançada. Em sistemas *para trás* ("backward"), as implicações usadas como regras-B operam numa base de dados global de metas até que uma condição de terminação envolvendo os fatos seja alcançada. Combinar operação "forward" e "backward" também é possível.

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.

5.2. Um Sistema de Dedução "Forward"

5.2.1. A forma AND/OR para Expressões de Fatos

O sistema "forward" 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 AND/OR*. Para converter uma fórmula na forma AND/OR, os símbolos "$\rightarrow$" (se existirem) são eliminados, usando a equivalência de ($W1 \rightarrow W2$) e ($\neg W1 \vee W2$). (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.

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 AND/OR consiste de subexpressões de literais conectados por símbolos "∧" e "∨". Note que uma expressão na forma AND/OR não está na forma de cláusula. Está muito mais perto da expressão original.

5.2.2. Usando Grafos AND/OR para Representar Expressões de Fatos

Um grafo AND/OR pode ser usado para representar uma expressão fato na forma AND/OR. Por exemplo, a árvore AND/OR da figura 1 representa a expressão fato posta na forma AND/OR acima. Cada subexpressão da expressão fato é representada por um nó no grafo. As subexpressões relacionadas disjuntivamente, $E_1,...,E_k$, de um fato, $(E_1 \vee ... \vee E_k)$, 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, $E_1,...,E_n$, de uma expressão, $(E_1 \wedge ... \wedge E_n)$, é representada por um único nó descendente conectado ao nó pai por um conector- l.

```
    -------------------------------------
  |Q(w,a) ∧ ((¬R(v) ∧ ¬P(v)) ∨ ¬S(a,v))|
    -------------------------------------
                /     \
               /       \
              ↙         ↘
     --------       ---------------------------
     |Q(w,a)|       |(¬R(v) ∧ ¬P(v)) ∨ ¬S(a,v) |
     --------       ---------------------------
                               /∪\
                              /   \
                             ↙     ↘
             ----------------    ---------
             |¬R(v) ∧ ¬P(v) |    |¬S(a,v)|
             ----------------    ---------
                    /  \
                   /    \
                  ↙      ↘
          -------       -------
          |¬R(v)|       |¬P(v)|
          -------       -------
```

Fig. 1. Uma representação em árvore AND/OR de uma expressão fato.

Os nós folhas da representação de grafo AND/OR 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*. Ele não tem nenhum ancestral no grafo.

Uma propriedade interessante da representação de grafo AND/OR 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 AND/OR. Então, as cláusulas que resultam da expressão $Q(w,a) \wedge ((\neg R(v) \wedge \neg P(v)) \vee \neg S(a,v))$ são:

$Q(w,a)$
$\neg S(a,v) \vee \neg R(v)$
$\neg S(a,v) \vee \neg P(v)$

5.2.3. Usando Regras para Transformar Grafos AND/OR

As regras de produção usadas pelo sistema de produção "forward" são aplicadas a estruturas de grafos AND/OR para produzir estruturas de grafos transformadas. Estas regras são baseadas em fórmulas implicacionais que representam conhecimento assercional 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 AND/OR), 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.

A restrição a antecedentes de literal único simplifica consideravelmente o processo de casamento na aplicação de regras aos grafos AND/OR. Por exemplo, pode-se transformar a implicação $(L1 \vee L2) \rightarrow W$ em um par de regras $L1 \rightarrow W$ e $L2 \rightarrow W$.

5.2.4. Usando a Fórmula Meta para Terminação

O objetivo do sistema de produção "forward" descrito é provar alguma fórmula meta a partir de uma fórmula fato e de um conjunto de regras. Este sistema "forward" 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 AND/OR. 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 arcos de casamento. O sistema de produção termina de forma bem sucedida quando ele produz um grafo AND/OR contendo um grafo de solução que termina em nós metas. (Na terminação, o sistema inferiu uma cláusula idêntica a alguma subparte da cláusula meta.)

5.3. Um Sistema de Dedução "Backward"

Uma propriedade importante da lógica é a dualidade entre asserções e metas em sistemas de prova de teoremas. Já foi visto uma instância deste princípio de dualidade nos sistemas de refutação por resolução. Lá a fórmula meta era negada, convertida na forma de cláusula, e adicionada às asserções em forma de cláusulas também. A dualidade entre asserções e metas permite que a meta negada seja tratada como se fosse uma asserção. Os sistemas de refutação por resolução aplicam resolução ao conjunto de cláusulas combinadas até que a cláusula vazia (denotando F) seja produzida.

Pode-se também descrever um sistema de resolução dual que opera nas expressões metas. Para preparar as fórmulas para tal sistema, deve-se primeiro negar a fórmula representando as asserções, converter esta fórmula negada ao dual da fórmula de cláusula (uma disjunção de conjunções de literais), e adicionar estas cláusulas a forma de cláusula dual da fórmula meta. O sistema deve então aplicar uma versão dual da resolução até que a cláusula vazia (agora denotando V) seja produzida.

Pode-se também imaginar sistemas mistos nos quais três formas diferentes de resolução são usadas, a resolução entre asserções, a resolução entre expressões metas e a resolução entre uma asserção e uma meta. O sistema "forward" descrito no último item deveria ser apontado como um destes sistemas mistos porque ele envolve o casamento de um literal fato no grafo AND/OR com um literal meta. O sistema de produção "backward", descrito aqui, é também um sistema misto que é, de alguma forma, dual ao sistema "forward". Sua operação envolve os mesmos tipos de representações e mecanismos que são usados no sistema "forward".

5.3.1. Expressões Metas na Forma AND/OR

O sistema "backward" é capaz de tratar de expressões metas de forma arbitrária. Primeiro, converte-se a fórmula meta na forma AND/OR pelo mesmo tipo de processo usado para converter uma expressão fato. Elimina-se os símbolos $\rightarrow$, move-se os símbolos de negação para dentro, Skolemiza-se as variáveis *universais*, e elimina-se os quantificadores existenciais. As variáveis restantes na forma AND/OR de uma expressão meta têm assumida a quantificação existencial.

Por exemplo, a expressão meta:

$$\exists y \forall x\ (P(x) \rightarrow (Q(x,y) \wedge \neg(R(x) \wedge S(y))))$$

é convertida para

$\neg P(f(y)) \vee (Q(f(y),y) \wedge (\neg R(f(y)) \vee \neg S(y)))$,

onde f(y) é uma função de Skolem.

A padronização das variáveis nas disjunções (principais) da meta leva a:

$\neg P(f(z)) \vee (Q(f(y),y) \wedge (\neg R(f(y)) \vee \neg S(y)))$.

(Note que a variável *y* não pode ser renomeada *dentro* da subexpressão disjuntiva.)

As fórmulas metas na forma AND/OR podem ser representadas como grafos AND/OR. Mas com expressões metas, os conectores-k nestes grafos são usados para separar *conjuntivamente* subexpressões relacionadas. A representação do grafo AND/OR para a fórmula meta exemplo acima é mostrada na figura 2. Os nós folhas deste grafo são rotulados pelos literais da expressão meta. Nos grafos metas AND/OR, chama-se qualquer descendente do nó raiz de *nó submeta*. As expressões que rotulam tais nós descendentes são chamadas de *submetas*.

```
    -----------------------------------------
   |¬P(f(z)) ∨ (Q(f(y),y) ∧ (¬R(f(y)) ∨ ¬S(y)))|
    -----------------------------------------
        /  \
       /    \
      ↙      ↘
 ----------    ------------------------------
|¬P(f(z))|    |Q(f(y),y) ∧ (¬R(f(y)) ∨ ¬S(y))|
 ----------     ------------------------------
                        /∪ \
                       /    \
                      ↙      ↘
               -----------    -----------------
              |Q(f(y),y)|    |¬R(f(y)) ∨ ¬S(y)|
               -----------     -----------------
                                   /  \
                                  /    \
                                 ↙      ↘
                          ----------   -------
                         |¬R(f(y))|  |¬S(y)|
                          ----------   -------
```

Fig. 2. Uma representação de grafo AND/OR de uma fórmula meta.

O conjunto de cláusulas na representação de forma de cláusula desta fórmula meta pode ser lida a partir do conjunto de grafos solução terminando em nós folhas:

$\neg P(f(z))$

$Q(f(y),y) \wedge \neg R(f(y))$
$Q(f(y),y) \wedge \neg S(y)$

As cláusulas metas são *conjunções* de literais e a disjunção destas cláusulas é a forma de cláusula da fórmula meta.

5.3.2. Aplicando Regras num Sistema "Backward"

As regras-B para este sistema são baseadas em implicações assercionais. Elas são asserções assim como são as regras-F no sistema "forward". Agora, entretanto, vai-se restringir as regras-B a expressões da forma

$$W \rightarrow L,$$

onde W é qualquer fórmula (assumida estar na forma AND/OR), L é um literal, e o escopo da quantificação de quaisquer variáveis na implicação é a implicação inteira. (Novamente, a restrição de regras-B a implicações desta forma simplifica o casamento e não causa dificuldades práticas importantes. Também, uma implicação tal como $W \rightarrow (L1 \wedge L2)$ pode ser convertida a duas regras $W \rightarrow L1$ e $W \rightarrow L2$.)

Tal regra-B é aplicável a um grafo AND/OR representando uma fórmula meta se esse grafo contém um nó literal rotulado por L' que unifica com L. O resultado da aplicação da regra é adicionar um arco de casamento a partir do nó rotulado por L' para um nó do novo descendente rotulado por L. Este novo nó é o nó raiz da representação do grafo AND/OR de Wu onde *u* é o u.m.g. de L e L'. Este u.m.g. rotula o arco de casamento no grafo transformado.

5.3.3. A Condição de Terminação

As expressões fatos usadas pelo sistema "backward" são limitadas àquelas na forma de uma conjunção de literais. Tais expressões podem ser representadas como um conjunto de literais. Análogo ao sistema "forward", quando um literal fato casa com um literal rotulando um nó literal do grafo, um descendente correspondente *nó fato* pode ser adicionado ao grafo. Este nó fato é ligado ao nó literal da submeta casada por um arco de casamento rotulado pelo u.m.g.. O mesmo literal fato pode ser usado várias vezes (com variáveis diferentes em cada uso) para criar nós fatos múltiplos.

A condição para terminação bem sucedida para o sistema "backward" é que o grafo AND/OR contém um grafo de solução *consistente* terminando em nós fatos. Novamente, um grafo de solução consistente é aquele onde as substituições do arco de casamento têm uma composição unificadora.

5.4. "Resolvendo" dentro dos Grafos AND/OR

O sistema "backward" descrito não é capaz de provar expressões de metas válidas ou tautológicas como $(\neg P \vee P)$ a menos que ele possa provar $\neg P$ ou P separadamente. O sistema "forward" não pode reconhecer expressões fatos contraditórias como $(\neg P \wedge P)$. Com a finalidade de superar estas deficiências, os sistemas devem ser capazes de realizar inferências intrameta ou intrafato.

Vai-se descrever como certas inferências intrameta podem ser realizadas. Considere, por exemplo, as seguintes expressões usadas num sistema "backward":

Meta

$(P(x,y) \vee Q(x,y)) \wedge V(x,y)$

Regras

R1: $(R(v) \wedge S(u,b)) \rightarrow P(u,v)$
R2: $(\neg S(a,s) \wedge W(r)) \rightarrow Q(r,s)$

Fatos

$R(b) \wedge W(b) \wedge V(a,b) \wedge V(b,b)$

Depois de aplicar as regras R1 e R2, tem-se o grafo AND/OR mostrado na figura 3. Este grafo tem dois literais complementares cujos predicados unificam com u.m.g. $\{ x/a, y/b \}$. O procedimento *Resolução de Meta Restrita ("RGR" - Restricted Goal Resolution)* permite eliminar nós folha complementares numa disjunção, para efeito de simplificação do grafo A/O. Um outro procedimento possível, caso o grafo gerado não resolva o problema, é gerar um outro grafo, partindo da expressão obtida através da aplicação da propriedade distributiva.

5.5. Uma Combinação de Sistemas "Forward" e "Backward"

Ambos os sistemas de dedução baseados em regras, "forward" e "backward", têm limitações. O sistema "backward" pode manipular expressões metas de forma arbitrária mas está restrito a expressões fatos consistindo de conjunções de literais. O sistema "forward" pode manipular expressões fatos de forma arbitrária mas está restrito a expressões metas consistindo de disjunções de literais. Pode-se combinar os dois sistemas e tirar vantagens de cada um sem as limitações de ambos?

```
                                  ----------------------------
                                  |(P(x,y) ∨ Q(x,y)) ∧ V(x,y)|
                                  ----------------------------
                                        /∪ \
                                       /    \
                                      ↙      ↘
                        ----------------   ---------
                        |P(x,y) ∨ Q(x,y)|  | V(x,y) |
                        ----------------   ----------
                              /  \
                             /    \
                            ↙      ↘
                   --------        --------
                   |P(x,y)|        |Q(x,y)|
                   --------        --------
                    ||{x/u, y/v}        || {x/r, y/s}
                   ↓↓                   ↓↓
              --------                  --------
              |P(u,v)|                  |Q(r,s)|
              --------                  --------
               /∪ \                      /∪ \
              / R1 \                    / R2  \
             ↙      ↘                  ↙       ↘
        ------  --------  {x/a, y/b}  ---------  -----
        |R(y)|  |S(x,b)|⇐==========⇒|¬S(a,y)| |W(x)|
        ------  --------            ---------  -----
```

Fig. 3. Um grafo AND/OR com nós literais complementares.

5.6. Exercícios

1. A partir do fato

$$(L \lor B) \land C$$

e das regras

1. A → G2 ∧ P
2. B → A ∧ M
3. C → H ∨ (D ∧ E)
4. D → G1
5. E → N
6. L → ¬H

deduza a meta

$$G1 \lor G2$$

usando técnica "forward" e com o auxílio de grafo de conhecimento. Se necessário, use RGR.

2. Com o auxílio de um grafo de conhecimento mostre, usando técnica "backward", que os fatos apoiam a meta:

Meta:

$$(K \wedge L) \vee (M \wedge R)$$

Regras:

1. $J \vee H \rightarrow K$
2. $F1 \wedge P \rightarrow L$
3. $N \vee (Q \wedge \neg P) \rightarrow M$
4. $F2 \rightarrow H$
5. $F3 \rightarrow Q$
6. $F4 \wedge S \rightarrow R$
7. $F4 \rightarrow S$

Fatos:

$$F1 \wedge F2 \wedge F3 \wedge F4$$

Use RGR, se necessário.

3. Usando as seguintes regras

1. $P \wedge Q \rightarrow R$
2. $S \rightarrow T$
3. $T \rightarrow P$
4. $S \rightarrow V$
5. $V \rightarrow Q$

e técnica "forward", demonstre R a partir de S. Monte grafo de conhecimento. Em seguida, num estilo demonstração por refutação, reproduza os mesmos passos deixando claras as resoluções efetuadas.

4. Repita o exercício anterior usando técnica "backward".

5. Considere a seguinte base de dados

1. $\forall x \forall y\ ((H(x) \wedge D(y)) \rightarrow F(x,y))$
2. $\exists y(G(y) \wedge (\forall z(R(z) \rightarrow F(y,z))))$
3. $\forall y(G(y) \rightarrow D(y))$
4. $\forall x \forall y \forall z((F(x,y) \wedge F(y,z)) \rightarrow F(x,z))$

e com base nela, prove a seguinte sentença:

$$\forall x \forall z\ ((H(x) \wedge R(z)) \rightarrow F(x,z))$$

6. Considere o sistema

Fato:

S

Regras:

1. $R \rightarrow \neg P$
2. $\neg J \rightarrow R$
3. $S \rightarrow (\neg M \vee \neg T)$
4. $\neg M \rightarrow (\neg J \vee K)$
5. $\neg T \rightarrow \neg Q$
6. $\neg U \rightarrow \neg Q$
7. $N \rightarrow \neg U$
8. $K \rightarrow N \vee L$

Meta:

$$\neg P \vee \neg Q \vee L$$

Prove a meta utilizando técnica "forward" (encadeamento progressivo).

7. É possível resolver o problema anterior usando técnica "backward" (encadeamento regressivo)? No caso afirmativo, indique um grafo solução.

BIBLIOGRAFIA

- Nilsson, N. J. (1982). Principles of Artificial Intelligence. Springer-Verlag.