

# 线性递归 DataLog 程序优化算法\*

王家华 曹路 金祥意 姚天顺  
(东北大学信息科学与工程学院 沈阳 110005)

**摘要** 提出了线性齐次 DataLog 逻辑程序的概念, 并为该类程序设计了一个优化的求解算法。在此基础上提出了求解一般线性 DataLog 程序的优化算法。该算法利用带有约束条件的递归调用方法, 将线性 DataLog 程序求解问题变换成齐次程序求解问题。算法简单, 易于实现, 可应用于任何线性 DataLog 程序的求解。

**关键词** DataLog 逻辑程序, EDB 谓词, DB 谓词, 魔集算法, 线性齐次逻辑程序  
**分类号** TP 31.13

## The Optimized Algorithm for Linear Recursive DataLog Program

Wang Jiahua, Cao Lu, Jin Xiangyi, Yao Tianshun  
(Northeastern University)

**Abstract** First, a concept of linear homogeneous logic programs is presented. A specific optimizing algorithm is devised for solving the logic programs. Furthermore, a optimizing algorithm which solves a general linear DataLog logic program is drawn. With successive recursive calls and constraint conditions, the algorithm transforms the problem of solving linear DataLog logic programs into the problem of solving homogeneous programs. The algorithm is simple and easy to be implemented.

**Key words** DataLog logic program, EDB predicate, DB predicate, magic rewriting algorithm, linear homogeneous logic program

### 1 引言

知识库系统(演绎数据库系统)得到了广泛的研究<sup>[1-3]</sup>。逻辑程序求解的优化算法一直是知识库领域研究的核心问题, 一些优化算法已被提出<sup>[4-6]</sup>。鉴于知识处理的复杂性, 现有的优化算法仅适用于一些特殊的程序, 如左线性程序优化, 计数技术和传递闭包优化程序等。魔集规则重写技术仅适用于一类特殊程序, 正如其名称所意味的, 该算法难以理解与实现。因此, 寻求一种能处理一般 DataLog 线性程序的简单优化算法是十分必要的。

一个知识库 KB 是一个二元组  $KB = \langle F, R \rangle$ 。其中,  $F$  为基原子的集合, 即存储在数据库中的关系被称为 EDB 数据库;  $R$  为规则的集合, 其形式为

$$p \leftarrow q_1 \& q_2 \& \dots \& q_n$$

$p$  中的每一个变量至少出现在一个  $q_i$  中,  $1 \leq i \leq n$ 。谓词  $p$  称为规则头部, 规则的右侧称为规则体, 其中  $q_i$  称为子目标。与谓词  $p$  对应的关系为 DB 数据库, 它是用规则定义的, 谓词  $p$  本身称为 DB 谓词。本文仅讨论谓词中不含函数符号, 并且规则体中仅有一个 DB 谓词, 即线性递归 DataLog 程序的情形。对于任何递归的逻辑程序, 只要它有非空解, 则其必包含至少一个非递归的规则。我们仅对下述两个规则和一个查询目标组成的逻辑程序  $L$  进行讨论。

$$R1: p(X_1, \dots, X_k) \leftarrow q(X_1, \dots, X_n)$$

$$R2: p(X_1, \dots, X_k) \leftarrow r(Y_1, \dots, Y_m) \& p(Z_1, \dots, Z_k)$$

$$G1: p(a_1, \dots, a_l, X_{l+1}, \dots, X_k)$$

对于规则 R2, 有  $X_i = Y_j$  或  $X_i = Z_s$ , 其中  $1 \leq i \leq k, 1 \leq j \leq m, 1 \leq s \leq k$ 。查询目标 G1 中的  $a_1, \dots, a_l$  代表常数。这里假定所有的约束都出现在前  $l$  个参数位置。对于其他任何线性递归 DataLog 程序,

\* 1999 - 04 - 19 收稿, 1999 - 09 - 18 修回

都可归结为一组或多组上述形式的程序。

## 2 DB谓词的装饰与线性齐次

### DataLog逻辑程序

为了讨论线性齐次逻辑程序的概念,首先需要研究规则中DB谓词中变量的约束情况。

#### 2.1 DB谓词的约束与装饰

例1 设有逻辑程序L1:

$$R3: p(X, Y, Z): - q(X, Y, Z)$$

$$R4: p(X, Y, Z): - p(U, Y, Z) \& r(X, U, V)$$

$$G2: p(a, b, Z)$$

当把目标G2和规则R4的头部统一时,规则R4中的变量X和Y得到约束组a和b。此时规则R4头部谓词中变量X和Y分别取到约束值a和b,规则体中的谓词p(U, Y, Z)中的Y已取得约束值b。

以符号 $p^\alpha$ 表示DB谓词p的约束状态,其中 $\alpha$ 称为谓词p的装饰。 $\alpha$ 为字符b和f的序列,如果谓词p的某一变量是约束的,则与之对应的 $\alpha$ 中的分量为b,否则为f。对于例1中的头谓词和体谓词,分别有 $p^{bbf}$ 和 $p^{fbf}$ ,其中bbf与fbf称为谓词p的装饰。这两个装饰表明,此时头谓词对应一个一元关系(p关系的第3列),而体谓词对应一个二元关系(由p关系的第1列和第3列组成),头谓词和体中DB谓词不再对应同一关系。

#### 2.2 线性齐次逻辑程序

鉴于规则体中DB谓词可以出现在体中的任何位置,规则体中DB谓词的约束情况因其位置不同而变化,因此体中DB谓词的装饰有可能不同。例如交换规则R4中两个子目标的位置可以得到:

例2 逻辑程序L2:

$$R3: p(X, Y, Z): - q(X, Y, Z)$$

$$R5: p(X, Y, Z): - r(X, U, V) \& p(U, Y, Z)$$

$$G2: p(a, b, Z)$$

当把目标G2和规则R5头部统一时,头部谓词X和Y得到约束值a和b,规则体中DB谓词p的变量Y取得约束值b。当处理第1个子目标EDB谓词r时,此时X是受约束的,因此仅需其对应关系R的一个子集,即其第1个分量为a的那些元组。在处理完EDB谓词r之后,则变量U和V的值由R的上述子集合所确定。此时规则体内的DB谓词p的变量U也是受约束的。例2中的头部与体内DB谓词p的约束情形都为 $p^{bbf}$ ,它们对应同一关系。

为了明显地表示逻辑程序与目标统一之后的约

束情况,将装饰 $\alpha$ 前缀到谓词名称p的前边。对于例2有

$$bbf. p(X, Y, Z): - q(X, Y, Z)$$

$$bbf. p(X, Y, Z): - r(X, U, V) \&$$

$$bbf. p(U, Y, Z)$$

这种头部DB谓词与规则体内DB谓词装饰相同的DataLog逻辑程序,称为线性齐次递归逻辑程序。左线性逻辑程序和右线性逻辑程序都是线性齐次逻辑程序。

## 3 线性齐次逻辑程序优化求解

在线性齐次逻辑程序中,由于递归规则的头部谓词和规则体内DB谓词具有相同的约束模型,因此它们有同度关系,并有对应的属性。可利用迭代方式求解这种递归的逻辑程序,即前一轮计算的规则头部关系可作为下一轮规则体中DB谓词的已知关系参与运算,因为它们的度和属性是相同的。

#### 3.1 约束关系的定义与计算

逻辑程序中的查询目标是对逻辑程序的一次调用,如同普通程序设计中带有实参的过程名是对过程的调用一样。目标中的约束值构成了一个初始的约束关系。递归规则体中DB谓词的约束变元构成了一个约束关系,因此对递归规则的每次调用都产生约束关系新的元组。

例如规则R5和目标G2,有

$$p(X, Y, Z): - r(X, U, V) \& p(U, Y, Z)$$

$$p(a, b, Z)$$

令初始约束关系 $M_0 = \{ a, b \}$ ,DB谓词p的约束关系属性为变元U和Y,以 $M(U, Y)$ 表示约束关系。用目标G2对R5的第一次调用,得到由U和Y组成的关系,Y值直接由初始约束获得,U值由关系R的U分量中满足初始约束的那些值获得。这等价于关系 $M_0$ 和关系R的自然联接,即

$$M_1(U, Y) = \underset{U, Y}{(M_0(X, Y) * R(X, U, V))}$$

下一次调用可得

$$M_2(U, Y) = \underset{U, Y}{(M_1(X, Y) * R(X, U, V))}$$

于是得到约束关系的递归关系代数方程为

$$M(U, Y) = M_0(X, Y) \underset{U, Y}{(M(X, Y) * R(X, U, Y))}$$

对于更一般的逻辑程序L(假定它是齐次的),则有

$$M(Z_1, \dots, Z_i) =$$

$$M(X_1, \dots, X_l) \quad (M(X_1, \dots, X_l) * \\ z_1, \dots, z_l \\ R(Y_1, \dots, Y_m))$$

上述关系方程中, 变量  $Z_1, \dots, Z_l$  是规则体中 DB 谓词  $p$  的受约束变元, 方程右边第 1 个  $M(X_1, \dots, X_l)$  代表初始约束, 第 2 个  $M(X_1, \dots, X_l)$  表示前一轮已计算的约束关系, 其中变量  $X_1, \dots, X_l$  是头部谓词中受约束的变量。

### 3.2 线性齐次DataLog 程序的求解算法

为了描述解DataLog 程序L 的一般算法, 首先从求解一个具体的逻辑程序L2 开始。

#### 算法 1 简化魔集算法特例

第 1 步: 计算约束关系

$$M(X, Y) = \{ a, b \}$$

$$TM(X, Y) = M(X, Y)$$

/TM 为临时约束关系 /

While TM  $\Phi$ Do

$$TM(U, Y) = (TM(X, Y) * \\ u, Y \\ R(X, U, V))$$

$$TM = TM - M$$

/除去已发现的元组 /

$$M = M \quad TM$$

End Do

输出M

第 2 步: 计算由非递归规则 R3 定义的关系

$$P(X, Y, Z) = M(X, Y) * Q(X, Y, Z)$$

第 3 步: 计算递归规则 R5 定义的关系

$$TR(X, Y, Z) = P(X, Y, Z)$$

/TR 为临时结果关系 /

While TR  $\Phi$ Do

$$TR(X, Y, Z) = \\ x, y, z \\ (M(X, Y) * R(X, U, V) * \\ P(U, Y, Z))$$

$$TR = TR - P$$

$$P = P \quad TR$$

End Do

输出最终结果 P

对于更一般的线性逻辑程序L, 仍然假定它是齐次的, 其求解算法如下:

#### 算法 2 简化魔集算法

输入: 线性齐次DataLog 程序及其 EDB 关系

输出: 给定目标的查询结果

方法:

第 1 步: 计算其全部约束关系

$$M(X_1, \dots, X_l) = \{ a_1, \dots, a_l \}$$

$$TM = M$$

While TM  $\Phi$ Do

$$TM(Z_1, \dots, Z_l) =$$

$$(M(X_1, \dots, X_l) * R(Y_1, \dots, Y_m)) \\ z_1, \dots, z_l$$

$$TM = TM - M$$

$$M = M \quad TM$$

End Do

输出M

第 2 步: 计算结果关系

$$P(X_1, \dots, X_k) =$$

$$(M(X_1, \dots, X_l) * Q(X_1, \dots, X_n)) \\ x_1, \dots, x_k$$

$$\Delta TR = P$$

While TR  $\Phi$ Do

$$\Delta TR = (M(X_1, \dots, X_l) * \\ x_1, \dots, x_k \\ R(Y_1, \dots, Y_m) * \\ \Delta TR(Z_1, \dots, Z_k))$$

$$R(Y_1, \dots, Y_m) * \\ \Delta TR(Z_1, \dots, Z_k))$$

$$\Delta TR = \Delta TR - P$$

$$P = P \quad \Delta TR$$

$$P = P \quad \Delta TR$$

End Do

输出  $P = \sigma_F(P)$

上述算法中, 由于  $P$  的关系通常比较庞大, 所以应用了增量法求解其递归的关系代数方程。

## 4 线性DataLog 逻辑程序的优化

线性齐次DataLog 逻辑程序是线性DataLog 逻辑程序中一种特殊的类型。如果一个线性程序不是齐次的, 则可通过交换子目标以使 DB 谓词的子目标处于规则的末尾, 得到更多的约束变量。例如规则 R4 中体内 DB 谓词的约束模型为  $p^{bf}$ , 而在 R5 中却为  $p^{bf}$ 。通过程序的递归调用, 最终可得到一个齐次程序, 利用上节中简化的魔集算法求解。算法如下:

#### 算法 3 线性DataLog 程序优化算法

输入: 线性DataLog 逻辑程序L 及其 EDB 关系

输出: 线性逻辑程序的固定点

方法:

第 1 步: 将查询目标与递归规则头部统一, 得到头部约束模型  $p^{\alpha_1}$  和体内 DB 谓词约束模型  $p^{\alpha_2}$

if  $\alpha_1 = \alpha_2$

then 调用算法 2 计算, 将结果赋值给 R, 转第 3

步

```

else 使 DB 谓词位于规则体内最右位置
endif
第 2 步: 调用递归过程 Comp (L, g1, R)
第 3 步: 输出结果 R
Procedure Comp (L, g, R: out)
/L 为输入的逻辑程序及其 EDB 关系 /
/g 为查询目标 /
/R 为输出的结果关系 /
/用  $\alpha_M(X_1, \dots, X_{\alpha})$  表示目标中的约束事实 /
begin
1) 将目标 g 与规则的头统一
2) if  $\alpha = \alpha$ 
3) then 调用算法 2 计算, 将结果赋给 R
4) else  $R_1 = \frac{(\alpha_M(X_1, \dots, X_{\alpha})) * Q(X_1, \dots, X_n)}{X_1, \dots, X_k}$ 
5) Comp (L, g2, R2)
6)  $R_0 = \frac{(\alpha_M(X_1, \dots, X_{\alpha})) * R(Y_1, \dots, Y_m) * R_2(X_1, \dots, X_k)}{X_1, \dots, X_k}$ 
7) R = R1 R0
8) endif
end

```

算法的第 1 步是对已是齐次的逻辑程序直接求解。如果不是齐次方程, 则交换规则体内子目标的位置, 使 DB 子目标位于规则末尾, 以取得附加约束变元。第 2 步调用递归过程 Comp。通常情况下, 每次调用时谓词的装饰将改变, 如不改变, 则表明已得到一个齐次程序。由于 DB 谓词  $p$  的变元个数是有限的, 所以经过有限次的调用之后, 最后一定能得到一个可以求解的齐次程序。实际上经过几次递归调用就可得到齐次程序。过程中的 4) 计算本次迭代中规则  $R_1$  的结果; 5) 计算在新的约束下体内 DB 谓词所对应的关系; 6) 计算本次迭代中规则  $R_2$  的结果; 7) 计算本次迭代中逻辑程序的结果。

## 5 结 语

本文提出齐次逻辑程序的概念, 并提出逻辑程

序优化算法。该算法不同于左线性、右线性优化算法以及魔集算法, 它可应用于求解任何线性 DataLog 逻辑程序。尽管我们论证了通过有限次调用, 可将线性逻辑程序求解转换成线性齐次程序求解, 但仍需进一步证明所需要的准确的调用次数。进一步的研究可将本文方法扩展成求一般的线性逻辑程序优化算法。

## 参 考 文 献

- 1 Francois Bancilhon. Naive evaluation of recursively defined relations In: On Knowledge Base Management Systems—Integrating Database and AI Systems New York: Springer-Verlag, 1985
- 2 Shamim Naqvi, Shalom Tsur. A logical language for data and knowledge bases In: Principles of Computer Science New York: Computer Science Press, 1989
- 3 王家华, 金祥意, 张宏霞, 等. 微机关系数据库的递归查询接口. 控制与决策, 1998, 13(2): 173-176
- 4 Catriel Beeri, Raghu Ramakrishnan. On the power of magic In: Proc of the ACM Symp on Principles of Database Systems San Diego, 1987. 269-283
- 5 王家华, 金祥意, 张宏霞, 等. 一类递归查询问题的快速求解算法. 控制与决策, 1999, 14(2): 140-144
- 6 Jeffrey D Ullman. Principles of database and knowledge base systems New York: Computer Science Press, 1989

## 作 者 简 介

王家华 男, 1944 年生。1970 年毕业于吉林大学, 1980 年获东北大学硕士学位, 现为东北大学信息科学与工程学院副教授, 主要从事知识库技术的研究。

曹路 男, 1975 年生。1997 年毕业于东北大学, 现为东北大学信息科学与工程学院硕士研究生, 从事知识库技术研究。

金祥意 男, 1971 年生。1993 年毕业于吉林大学, 现为东北大学信息科学与工程学院硕士研究生, 从事知识库技术研究。

姚天顺 男, 1934 年生。1956 年毕业于吉林大学, 现为东北大学信息科学与工程学院教授, 博士生导师, 主要从事计算机科学理论的研究。