知识库系统KBASE中的规则规范化理论*

马学强 施伯乐

(复旦大学)

RULE NORMALIZATION THEORY IN KNOWLEDGE BASE SYSTEM KBASE

Ma Xueqiang and Shi Baile

(*Fudan University*)

ABSTRACT

In this paper we provides an overview of KBASE — a knowledge base system supported by a relational database management system. It follows a safe and efficient query-evaluation approach of compiling the query-patterns given by the user into Relational Algebraic Processing (RAP) trees. After the whole architecture and the inferrence mechanisms of KBASE is presented, we describes thoroughly the rule normalization technique used in KBASE. Three kinds of normal form are clarified. The transformation algorithms and good properties of each kind of normal form are also given.

摘 要

本文对一个建立在关系数据库INGRES上的知识库系统—KBASE作了简要介绍。该系统采用由用户给出查询模式，系统将查询模式编译成关系代数操作树(RAP树)的新而高效的查询计值方法。在介绍了整个系统的结构和推理模型的基础上，文章着重讨论了规则规范化理论，提出了三种范式的定义、转换方法及优良性质。

§1. 引 言

* 1989年10月11日收到，1989年12月15日定稿。

自 Kowalski 等人给出 Horn 子句的不动点语义和操作语义后，逻辑便成为一种编程语言。逻辑程序设计语言与演绎数据库的紧密结合，产生了知识库技术。知识库系统是用来存贮并管理大量格式化数据和大量演绎规则的。它的操作对象是实关系、虚关系和规则。规则的大量性和可操作性应是知识库系统区别于演绎数据库系统的重要特征。

知识库系统提供给用户的界面一般是一种基于逻辑推理的逻辑程序设计语言。而其内部处理机制通常有两种不同的途径。一种是以 PROLOG 为典型代表的基于证明论的方法，这种方法固有的低效性决定了它不适用于处理大量数据和知识的领域。另一种是基于模型理论的自底向上方法。这种方法的计算模型是“每次一个集合”的，它充分利用了关系数据库系统在管理、操作大量数据方面的现有技术(如高效的联结算法，模式修改设施等)。近几年来，由底向上方法的研究取得了令人瞩目的进展，出现了如 HN 法(4) 魔计数法等一些实用的优化方法。

KBASE 是一个正在研制的在数据库系统支持下的知识库系统。它采用并发展了国际上的编译—优化实现方法，是一种基于模型论的方法。下面一节将介绍 KBASE 系统概貌，第三节介绍 KBASE 系统中的规则规范化理论。

§2. KBASE 系统概貌

KBASE 系统建立在 MICRO-VAX/VMS 的关系数据库系统 INGRES 上，其核心语言 KBASE 是一种支持否定，集合，递归，复杂项等机制的逻辑程序语言。它采用先按照用户给出的查询模式进行优化，然后编译成关系代数的实现方法。这种由底向上的实现机制提供了较好的非过程性和动态查询效率，适合于大量数据和大量规则的知识处理领域，且为建立更加高级的接口，如模糊逻辑模型和时态逻辑模型提供了基础框架。KBASE 还可作为知识密集场合的应用系统，如专家系统等的原型和核心。

KBASE 源程序由规则，事实，查询模式和查询四部分组成。KBASA 中的谓词名相应于 INGRES 中的关系框架名，谓词中的域可以赋名，而不必象 PROLOG 那样用位置来表征。

查询模式是对给定谓词所进行查询的格式。设 p 是一个 n 维谓词，则 p 的一个合法的查询模式为：$p(\delta_1, \delta_2, \ldots, \delta_n)$　$\delta_i \in \{!, ?, -\}$ $(i = 1, 2, \cdots, n)$

其中标! 的参量为该类查询中已被约束的参量，标? 的参量为未被约束的待求值参量，标— 的参量为该类查询中不涉及到的参量。

例如，设原子 supply (supplier, part, project, quantity) 表示供应商 supplier 向项目 project 提供数量为 quantity 的零件 part。若用户给出查询模式 supply(! —, ? , —), 则表示用户需要进行给定供应商，要得到他所供应零件的项目这样一类查询。因此查询? supply('john', —, x, —) 为该查询模式的查询。用户如需进行某类查询，必须首先向系统提供相应的查询模式。

KBASE系统可分成四大模块，即语法分析器，优化器，编译器，DB接口器。系统的数据流图见图1。为提高系统效率，系统建立下面三种中间数据结构：规则表，查询模式索引表和关系代数操作树(简称为RAP树)。

语法分析器完成下面两个任务：

(1). 将用户输入的各条规则转化为内部表示——规则表。并提供用户对已输入的规则进行编辑的设施。

(2). 在INGRES系统的支持下，将KBASE程序中的事实存入数据库，同时，用户还可对数据库中的事实进行更新及对关系模式进行修改。

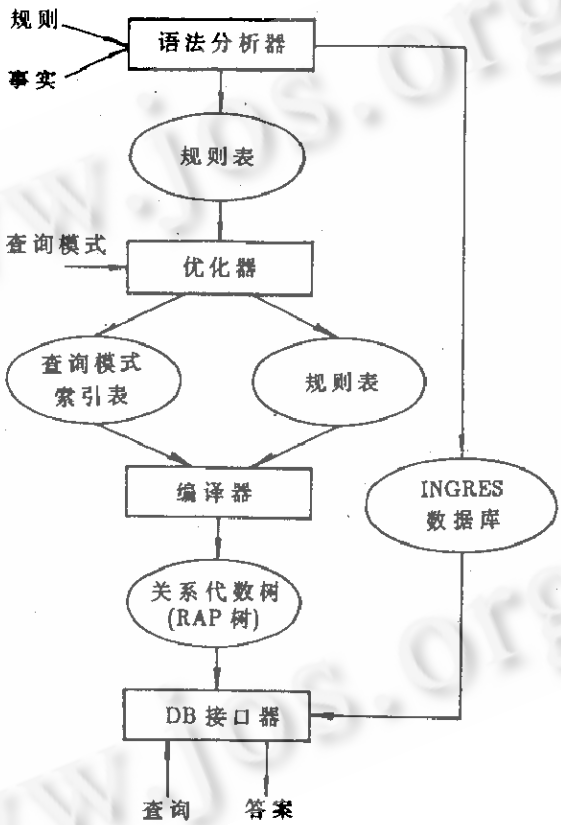

图1 KBASE系统数据流图

优化器根据用户给出的查询模式产生查询模式索引表，并对规则表中的规则进行规范和优化，将其转换成一个保证终止且计算高效的规则集。

查询模式索引表指明了每类查询模式计值的RAP树。RAP树中每个结点包含运算符域(关系或，笛卡尔积，关系差)，选择条件域，投影范围域和指向各棵子树的指针域。RAP树中的每个结点q都对应一个关系r(q)。叶结点的关系是相应的数据库基关系。设某非叶结点p的运算符域取值θ，选择条件域取值F，投影范围域取值d，则有：$r(p)=\sigma_F(\pi d(\theta(r(p_1),r(p_2,\ldots,r(p_n)))))$.

RAP 树指针域用以指向对该类查询模式计值的RAP 树的根结点。编译器的主要功能是根据查询模式表和规则表，生成对查询计值的RAP 树。编译结束后，查询模式索引表中的RAP 树指针即指向编译产生的相应RAP 树根结点。

数据库接口器对给定的查询，在索引表中寻找与之匹配的查询模式，对相应的RAP 树中的已知量进行代真，然后将该RAP 树转化为一系列INGRES 操作，把执行结果显示给用户。

§3. 规则规范化理论

优化器中的优化可分成两步：第一步称为规范化，它将用户输入的规则规范化成某种特定的范式。第二步称为重写优化，它针对用户给出的每一种查询模式，将原来的规则改写成保证终止且计算高效的新规则。

规范化方法是关系数据库设计理论特有的一种技术，它将用户给出的原始模式转换成满足某种良好性质的新的模式(范式)，以减少数据冗余和消除更新异常。下面我们对Horn 子句逐级约束，得到三种具有良好性质的范式。

首先我们来考虑规则体中的否定问题。一个带否定的KBASE 程序所对应的定义在其Herbrand 域上的映射关系不满足单调性，相应的递归方程可能无最小不动点或存在多个最小不动点，因而无法用不动点来刻划其语义。若一个KBASE 程序LP 的Herbrand 域H 可分割成不同的层次 $H_1, H_2, \cdots$(不妨称i 为Hi 的层次)，使得对于LP 中的每一条规则，体中的负文字对应的层数严格小于规则首部所对应的层数，而体中的正文字对应的层数不大于规则首部所对应的层数。这样的话，LP 就称为是可分层的。

下面的定义中，我们将不是 $x\theta y$ 形式的正文字(θ 为 $=, <, \leq, >, \geq, \neq$) 称为普通正文字。

定义1.　若规则r 为下列形式

$P: -q_1, q_2, \ldots, q_n$，其中p 为一个正文字，$q_i$ 为一个文字$(i = 1, 2, \ldots, n)$，且满足

(1) 可分层，(2) 若r 的体部出现负文字，则其参量中的所有变量均在体部中某个普通正文字的参量中出现。

(3) 若r 的体部出现x=y 形式的目标，则x, y 中至少有一个应在体部中某个普通正文字的参量中出现，(4) 若r 的体部出现 $x\theta y$(θ 为 $<, \leq, >, \geq$ 或 $\neq$) 形式的目标，则x, y 都应在体部中某个普通正文字的参量中出现，

则称r 为第一范式，记作1NF。若一个KBASE 程序LP 中的所有规则均为1NF，则称LP 为1NF。

对一个KBASE 程序，若对每个合法查询计值时都能保证终止且不会产生无限中间关系，则称此KBASE 程序为安全的。

定理1: 一个1NF 的KBASE 程序是安全的。

KBASE 对用户输入的规则进行检查，若不满足1NF 则给出错误信息。

定义2:　若规则r 为1NF 且满足下面两个条件:

(1) 规则中所有变量都不仅出现一次(2) 规则头谓词中的所有参量为两两不相同的变量, 则称r为第二范式, 记作2NF。若一个KBASE程序LP中所有规则均为2NF, 则称LP为2NF。

显然, 满足2NF的规则中, 规则头谓词中的所有变量均在规则体中出现。

为了进一步说明2NF的优良特性, 我们需引入"可匹配"的概念并分析它和可合一之间的区别和联系。

定义3: 一个类型原子为如下形式

$q(a_1, a_2, \ldots, a_n)$, 其中$a_i \in \{c, v, f\}(i = 1, 2, \ldots, n)$

任何一个KBASE原子或查询模式都可经下列步骤转化成一个类型原子:

将参量中出现的所有常量和'!'用'c'替代, 所有变量、'?'和'—'用'v'替代, 所有表达式用'f'替代。

若某个KBASE原子(或查询模式) p可转化为一个类型原子q, 则称p满足q。

定义4: 给定两个类型原子q_1, q_2,

(1) 若对所有满足q_1的KBASE原子(或查询模式) p_1和所有满足q_2的KBASE原子(或查询模式) p_2, p_1和p_2总可合一, 则称q_1和q_2可匹配。记作$T(q_1, q_2) = m$, 其中T称为匹配函数。

(2) 若对所有满足q_1的KBASE原子(或查询模式) p_1和所有满足q_2的KBASE原子(或查询模式) p_2, p_1和p_2总不可合一, 则称q_1和q_2不可合一。记作$T(q_1, q_2) = n$。

(3) 若q_1, q_2即不满足(1)也不满足(2), 则称q_1和q_2可不完全合一。记作$T(q_1, q_2) = u$。

显然, 当两个类型原子的谓词类型不同时, 它们的匹配函数等于n。当它们的谓词类型相同时, 其匹配函数取决于各自的变量类型。与定义类型原子间的匹配函数类似, 我们可在谓词参量间定义匹配函数$t(a_1, a_2)$, a_1, a_2是c, v, f三者之一, 其值域是$\{m, u, n\}$, m, u, n的含义同前。

图2的表给出了不同类型参量间的匹配函数t的值。

我们可进一步在匹配函数值域$\{m, u, n\}$上定义运算*, 其运算规则见图3。*运算满足可结合律。

t	c	v	f
c	u	m	n
v	m	m	m
f	n	m	u

图2 匹配函数表

*	m	u	n
m	m	u	n
u	u	u	n
n	n	n	n

图3 *运算规则表

现在我们可以定义两个具有相同谓词类型的类型原子$p(a_1, a_2 \ldots, a_n)$和$p(b_1, b_2, \ldots, b_n)$的匹配函数如下:

$$T(p(a_1, a_2, \ldots, a_n), p(b_1, b_2, \ldots, b_n)) = t(a_1, b_1) * (a_2, b_2) * \ldots * t(a_n, b_n)$$

引理1:

若类型原子q的所有参量均取v,则对任意具有相同谓词类型的类型原子q',成立　　$T(q,q')=m$

略证: 设 $q=p(v,v,\ldots,v), q'=p(b_1,b_2,\ldots,b_n)$

由匹配函数定义知 $T(q,q')=\prod_{i=1}^{n}t(v,b_i)$, 由图2的匹配函数知 $t(v,b_i)=m(i=1,2,\ldots,n)$, 再由图3的运算规则知 $\prod_{i=1}^{n}t(v,b_i)=m$. 证毕。

定义5: 给定一组规则集R和一个查询(或查询模式)Q,若在此查询的计值过程中产生的每个目标总能与任一和它具有相同谓词类型的规则首部可匹配,则称R对于Q是可静态合一的。

由引理1不难得到如下定理

定理2: 任一个2NF规则集对于所有查询模式是可静态合一的。

对于KBASE这样一个基于查询模式的计值系统,在编译前将规则转化为2NF的目的就是在未给出具体的查询时就能静态决定一个目标能否与一条规则的首部合一成功。定理2为KBASE系统采用静态编译方法奠定了理论基础。

算法1　2NF规范化算法

输入: 一组1NF规则

输出: 给出一组信息等价的2NF规则或给出无法规范化的信息。

步骤: 对于输入规则集中每一条1NF规则r:

$p(\alpha_1,\alpha_2,\ldots,\alpha_n):-\text{body}$

其中body为 $\theta_1q_1(\beta_h,\ldots,\beta_{1k_1}),\ldots,\theta_hq_h(\beta_{h1},\ldots,\beta_{hk_h})$, θ_i 为谓词前的正负符号($i=1,2,\ldots,h$)。并记dom(x)为变量x可取的值域。

分别执行下列规范化步骤:

(1). 对 $\alpha_1,\ldots,\alpha_n$ 中出现的所有常量或表达式 α_i, 执行下列步骤:

(a) 产生一个新的变量 α_i',

(b) 将r改写为　　$p(\alpha_1,\ldots,\alpha_{i-1},\alpha_i',\alpha_{i+1},\ldots,\alpha_n):-body,\alpha_i'=\alpha_i$

(2). 对 $\alpha_1,\ldots,\alpha_n$ 中任意一对有相同名字的变量 α_i,α_j, 执行下列步骤:

(a) 产生一个新的变量 α_i',

(b) 将r改写为　$p(\alpha_1,\ldots,\alpha_{i-1},\alpha_i',\alpha_{i+1},\ldots,\alpha_n):-body,\alpha_i'=\alpha_j$

(3). 对 $\alpha_1,\ldots,\alpha_n$ 中任一不在规则体部出现的参量 α_i, 执行下列步骤:

若 $dom(\alpha_i)$ 为一有限集, 则将r改写成: $p(\alpha_1,\alpha_2,\ldots,\alpha_n):-body,dom(\alpha_i)$

否则就将此规则标记为不可转化为2NF。

(4) 对r中仅出现一次的变量, 用符号'—'代替。

执行完(1), (2), (3), (4)后, 若存在一条规则被标记为不可转化为2NF, 则返回无法规范化信息。

定理3:

(1). 算法1成功返回的充要条件是输入规则集的每条规则中仅出现一次的变量的可取值域有限。

(2). 若算法1成功返回, 则返回的规则集为与输入的规则集信息等价的2NF规则集。

下面我们来考虑递归问题。互相递归关系是一种等价关系，我们可按照互相递归关系将所有递归关系将所有递归谓词划分成一个个等价类，其中的每个等价类称为一个递归族。

定义6

若一个KBASE程序LP中的每个递归族中仅含单个递归谓词，则称LP满足第三范式，记作3NF。由前面一节可知，RAP树是按照查询模式索引表和规则表来构造的。索引表中每行相应于一个或结点，规则表中每行相应于一个与结点。由于每个递归族中的所有递归谓词必须同时求值，当一个递归族中包含很多谓词时，每对其中一个递归谓词求不动点，势必影响到RAP树中很多的结点，造成运算的复杂性。因此，将KBASE程序转化为3NF的目的就是将不动点求值涉及到的RAP树结点范围尽量压缩。

定理4:

对3NF的KBASE程序的递归谓词求不动点时，运算牵涉到的结点范围最小。

算法2 3NF规范化算法

输入: 一组2NF规则

输出: 给出一组信息等价的3NF规则或给出无法规范化的信息。

步骤: 将输入的规则集按相互递归关系划分成一个个递归族，然后对每个递归族$\{p_1,p_2,\ldots,p_m\}$执行下列步骤:

(1) $j\leftarrow 1$ (2) $i\leftarrow 1$

(3) 在所有以p_i为首部的规则中，将其体部出现的同族递归谓词$p_k(k=1,2,\ldots,i-1,i+1,\ldots,m)$,用其定义式中和各条规则来替代。

(4) $i\leftarrow i+1$, 若i大于m, 则转至(5), 否则转至(3)。

(5) $j\leftarrow j+1$, 若j大于m则退出循环，否则对经过变换后的规则集重新执行步骤(2), (3), (4)。

若变换后的每条以p为首部的规则中，其体部不出现除p外的同一递归族中的谓词，则返回成功信息和转换后的规则集，否则返回无法规范化的信息。

例如，对于下面的规则集(其中A, B, 均为谓词，不考虑出口规则)。

$P:-Q,B$

$Q:-P,A$

执行算法2后得到如下的3NF规则集:

$P:-P,A,B$

$Q:-Q,B,A$

又如，对于下面的规则集(A, B, C, D为基谓词)

$P:-A$

$P:-Q,P,B$

$Q:-C$

$Q:-P,Q,D$

算法2将给出无法规范化的信息。实际上，对上面递归族$\{P,Q\}$的不动点计值，相当于一个求含闭包运算的正规式的闭包，运算复杂度较高。遇到这种情况，可采用类似于数值分析中的Jacobi算法和Gauss-Seidel算法来解。

定理 4:

若算法 2 成功返回，则返回的规则集为与输入的规则集信息等价的 3NF 规则集。

§4. 小结

查询优化是现在知识库系统研究中一个异常活跃的研究方向。美国的 MCC 组织，日本的 ICOT 机构，欧洲共同体的 Esprit 项目组都在从事这方面的研究，并取得了不少进展。F. Bancilhon 等人在 86 年给出了魔集法，计数法等逻辑程序查询优化方法。87 年 MCC 的 Sacca 和 Zaniolo 又将魔集法的安全性与计数法的高效性结合起来，提出了魔计数法。但这些优化方法都依赖于用户所给查询中的约束条件，因此重写优化必须在读入用户查询后才能进行，影响了效率。

本文提出，查询优化可分成两个部分，第一部分是不依赖于查询的优化，第二部分是依赖于查询的优化。第一部分实际上是规范化过程。本文提出了逻辑数据模型的“范式”的思想，并分别给出了三种范式的定义，转换方法及优良性质。对第二部分的处理，本文提出了按照用户给定的查询模式进行重写优化的设想，使优化和编译工作能在用户查询之前静态完成。

本文的思想为知识库系统中采用重写优化技术建立了一种新的更有效的统一框架。KBASE 原型上的初步测试结果表明，本文的方法比之直接的动态重写优化，用户查询的响应时间可缩短一个数量级。

参考文献

1. Beeri, C., Kanellakis, P., Bancihon, F., Ramakrishnan, R., “Bounds on the Propagation of Selection into Logic Programs”, Proceedings of the 6th PODS, ACM, 1987.
2. Ceri, S., Gottlob, G., Lavazza, L., “Translation and Optimization Logic Queries: the Algebraic Approach”, Proceedings of the 12th VLDB, 1986.
3. Gray, P. M. D., Moffat, D. S., paton, N.W., “A Prolog Interface to a Functional Data Model Database”, Proc. 1st EDBT Conference, 1988.
4. Henschen, L.J, Naqvi, S.A, “On Compiling Queries in Recursive First-order Databases”, JACM 31, 1, 1984.