第 12 卷第 2 期　　广 西 科 学 院 学 报　　Vol. 12, No. 2
1996 年 6 月　　Journal of Guangxi Academy of Sciences　　June 1996

知识库中的空值初探
The Primary Research of Null Value Processing on Knowledge Base Systems

袁冰艳　　华　为
Yuan Bingyan　　Hua Wei

（广西卫生管理干部学院　南宁　530021）
(Guangxi Health Manage Cadre Institute, Nanning, 530021)

（上海市电话局设计所　上海　200092）
(Shanghai Design Institute of Telephone Exchange, Shanghai, 200092)

摘要　分析了空值环境下的三种类型的查询策略，以及 Datalog 查询求值的 Semi-Naive 算法，给出了一个从子目标关系空值特性导出头关系空值特性的一种方法，使改进后的 Semi-Naive 算法能在带有空值的 EDB 数据库中对 Datalog 查询进行正确求值。

关键词　知识库　空值　Datalog 查询

Abstract　By analysing the three kinds of query strategies under null values environment and the Semi-Naive algorithm of Datalog query Evaluation. A method to obtain the null value features of IDB's attributs from that of EDB's is given for the Semi-Naive algorithm to correctly complete the computation in Datalog query of EDB with null values.

Key words　knowledge bases, null values, Datalog queries

当前，有关空值环境下关系数据库的查询处理方法已有诸多的研究，不仅定义了存在型空值、不存在型空值和占位型空值，还定义了空值模型和三值逻辑。知识库是关系数据库和人工智能技术相结合的产物，具体地讲是将逻辑程序技术引入了关系数据库中，那在逻辑程序中又如何处理空值问题呢？我们将从分析知识库中的查询求解的 Semi-Naive 算法来进行讨论。

1　基本概念

1.1　知识库的概念

知识库系统是具有以下两个特征的逻辑程序设计系统：

1996-03-25 收稿。

1.1.1　有一个既作为查询语言又作为宿主语言的描述性语言。

1.1.2　支持数据库系统的全部功能。

知识库具有演绎检索的功能，是通过将逻辑规则转换成关系代数操作而实现的。Datalog是知识库系统中的数据模型。

1.2　逻辑规则的语义

逻辑规则的含义有以下三种解释：

1.2.1　规则的证明论解释：即用给定的或已经证明的事实代替规则体，然后证明规则事实。

1.2.2　规则的模型论解释：模型是对规则为真的“解释”，“解释”是对规则中的谓词可能的实例赋于真值或假值。

1.2.3　规则的计算语义：如果要使逻辑规则变为计算机程序则必须要有执行程序的算法，算法的结果即为规则的语义。

1.3　Datalog查询求值方法分析

Datalog的递归程序由多条规则组成，一个规则由规则头和规则体组成，规则体由多个子目标关系组成，Datalog以两种方式定义关系：

1.3.1　EDB关系：存储在数据库的关系，对称的谓词称EDB谓词。

1.3.2　IDB关系：由逻辑规则定义的关系，对应的谓词称IDB谓词。

根据文献[3]，Datalog的查询求值方法是Semi-Naive算法。该算法求IDB谓词所对应的关系的方法是：

1.3.2.1　对每个以IDB谓词p_i为头的规则计算出体关系即对每一个子目标做选择、投影，然后做自然联结。

1.3.2.2　把自然联结的结果投影到p_i所对应的变量上，然后做并运算。

由于EDB谓词出现了空值，因此，IDB谓词的关系运算也需要采用空值的处理方法。

根据文献[1]，把占位型空值φ^-理解为φ^0（不存在型空值）和φ^*（存在型空值），这样对三种类型的空值都能进行处理了。又据文献[1]定义的definite，indefinite，maybe这三种查询操作类型，做出一个输出空值环境下的选择和自然联接运算结果的算法Null-Calculate.

算法：Null-Calculate（R，C，type）

输入：运算关系R，条件表达式F，运算符号C，操作类型Type

输出：运算结果的元组集合t

```
begin
  t:=φ
if c=∞ then R=(R1, R2)
case type of
  definite: for i:=1 to n do /*c=σ时，n为R元组数，C=∞时，n为R1与R2元组数之积*/
          if (F中有φ-) then (置为φ0)
          if (F的值为ture) then if C=σ then t:=t+ri; /*ri∈R*/
            else t:=t+ri; /*ri∈R1∞R2*/
          next i;
  indefinite: for i:=1 to n do
              if (F中有φ-) then (置为φ0)
```

```
                if (F 的值为 ture or maybe) then
                  if c=σ then t: =t+r_i; /* r_i∈R */
                    else t: =t+r_i; /* r_i∈R_1∞R_2 */
                next i;
  maybe: for i: =1 to n do
          if (F 中有 φ⁻) then (置为 φ*) /* 其语义范围为相应的属性值域 */
          if (F 的值为 ture or maybe) then
            if C=σ then t: =t+r_i; /* r_i∈R */
              else t: =t+r_i; /* r_i∈R_1∞R_2 */
      next i;
    endcase;
end;
```

我们用 Datalog 的查询求值 Semi-Naive 算法进行求值操作时，不能保证所有的操作都是作用于已知的 EDB 关系，IDB 关系常作为中间结果存放，由于引入了空值，因此，中间关系的建立及维护就非常重要，这时我们需要给出一个从子目标关系空值特性导出头关系空值特性的方法。

2　Null-Value-Property 算法

分析了 Semi-Naive 算法后，我们得出从 EDB 谓词空值特性导出 IDB 谓词特性的算法 Null-Values-Property：(type)

Null-Value-Property 算法.

输入：一组校正过的规则，EDB 关系 R_1，…，R_k，IDB 谓词 p_1，…，p_m，p_i 对应的表达式集合（m 个），集合元素为以 p_i 为规则头的规则体表达式，查询操作类型 type

输出：各 p_i 空值特性 AP_i（n）（设 p_i 有 n 个属性）

AP_i（x）=0　表示 p_i 第 x 个属性只能取常规值 r

AP_i（x）=1　表示 p_i 第 x 个属性只能取常规值及 φ^0

AP_i（x）=2　表示 p_i 第 x 个属性只能取常规值及 φ^*

AP_i（x）=3　表示 p_i 第 x 个属性只能取常规值及 φ^*、φ^0

```
begin /* EDB 空值特性 */
    for i: =1 to n do
     for j: =1 to R_i 属性个数  do
      ARi (j) =0;
    for i: =1 to n do
     for j: =1 to R_i 属性个数 do
     begin
      Fφ⁰: =0; Fφ*: =0; Fφ⁻: =0;
       if π$j (Ri) 有 φ⁰ the Fφ⁰: =1;
       if π$j (Ri) 有 φ* the Fφ*: =1;
       if π$j (Ri) 有 φ⁻ the Fφ⁻: =1;
         case type of
          definite, indefinite: if (Fφ⁰ or Fφ⁻) and Fφ* then ARi (j): =3;
          else if Fφ* then ARi (j): =2;
            else if (Fφ⁰ or Fφ⁻) then ARi (j): =1;
```

```
        maybe: if Fφ⁰ and (Fφ⁺ or Fφ⁻) then ARi (j) :=3;
               else if Fφ⁺ or Fφ⁻ then ARi (j): =2;
                 else if Fφᶜ then ARi (j): =1;
      end;
  for i: =1 to m do /* 按递归程序求 APi */
      for j: =1 to pi 属性个数 do APi (j) :=0;
       repeat
        for i: =1 to m do
         for j: =1 to pi 属性个数 do QPi (j): =APi (j);
          for i: =1 to m do
           for j: =1 to pi 属性个数 do
            APi (j): =Null-Value (attr);
      until QPi: =APi for all i, for all j
end;
算法: Null-Value (Attr)
    输入: 查询操作类型 type、pi 的一个属性名 Attr 及该 pi 相应的表达式集合 E
    输出: pi 的属性 Attr 的空值特性
begin
  result: =0
  for x: =1 to size of (E) (元素个数) do
   begin
    loc: =Attr 在第 x 个表达式的第几个子目标 S;
     K: =Attr 存在子目标 S 中的位置;
     temp: =AS (K);
     for y: =loc+1 to 第 x 个表达式的子目标数 do
      if Attr 出现在第 y 个子目标 SS 中,
        then
         begin
          K: Attr 与 SS 的 $K 名字相同;
          Cur: =ASS (K);
         Case type of
         definite: if (temp, cur 全 1)or (temp, cur 全 3) or (temp, cur, 一个为 1,一个为 3)
                    then temp: =1 else temp: =0
          indefinite, maybe: if (temp, cur, 一个为 0, 一个为 2) or (temp, cur, 一个 0, 一
个为 3)
                    then temp: =2;
                    else if (temp, cur, 一个为 1, 一个为 2)
                    the temp: =3;
                    else lemp: =min (temp, cur);
        end;
 if (result, temp, 一个为 1, 一个为 2)
     then result: 3;
     else result: =max (result, temp);
     end;
```

```
    return (result);
end;
```

3 Improve-Semi-Naive 算法

由于 Null-Value-Property 算法根据三种类型的查询策略而给出了从 EDB 关系空值特性导出 IDB 关系空值特性的方法，这使得 Datalog 的查询求解 Semi-Naive 算法能在带有空值的 EDB 数据库中对 Datalog 查询进行正确求值，以下是带有查询操作类型的 Semi-Naive 算法的改进算法。

算法：Improve-Semi-Naive (type)

输入：一组校正过的 Datalog 规则，EDB 关系 R_1、R_2、…、R_k，IDB 谓词 p_1，p_2、…、p_m，查询操作类型 type

输出：各 IDB 谓词对应关系的元组集合 S

```
begin
    for i: =I to m do begin
    △Pi: =Eval (pi, R1, …, Rk, φ…φ);
  / * Eval 算法参照文献 [3] 定义，此时若在关系运算中有选择或自然联接则应根据查询操作类型调用 Null-Calculate 算法 * /
    pi: =△Pi
  end;
  repeat
   for i: =I to m do
    △Qi: =△Pi
    for i: =1 to m do begin
    △Pi: =Eval-Incr (pi, R1, …, Rk, p1, …, pm; △Q1, …, △Qn);
/ * Eval-Cncr 算法参照文献 [3] 定义，此时，若在关系运算时遇到自然连结和选择运算时，则应根据查询操作类型调用 Null-Calculate 算法 * /
    △Pi: =△Pi=Pi
end ;
   for i: =I to m do
     pi: =pi U△Pi
  until △P1: =φ for all i
end;
out put pi's;
end;
```

4 结束语

本文给出了从 EDB 谓词导出 IDB 谓词空值特性的 Null-Value-Property 算法和 Datalog 查询求值的 Improve-Semi-Naive 算法，实现了空值环境下的知识库中的查询处理，也为进一步研究含空值关系的知识库的更新处理提供了工具。

（参考文献略）