完美模型的充要条件及其结构分析*

黄东斌 (广州市电信局科技处 广州 510055)
李 磊 (中山大学岭南学院软件研究所 广州 510275)

摘要 纯Datalog的唯一模型定理在Datalog-Not程序中不成立,因此,在Datalog-Not程序的所有Herbrand模型中,必须有一种模型的选择标准来确定程序的语义.完美模型关系是一种合理的选择标准.然而,由于完美模型的复杂性阻止了它的发展.本文分析了完美模型,通过有向图,给出了有完美模型的充要条件,并且给出构造算法及其正确性证明.本文得到的结果是:对任一自反有向图G,都能构造一个Datalog-Not程序P,使得P的完美模型图与G同构.

关键词 逻辑程序,Herbrand模型,分层程序,完美模型.

可以认为,Datalog是一阶谓词逻辑的一个子集,它在模型论上具有很好的性质,如模型唯一性和模型的交集仍是模型等.[1~4]但是,Datalog由于缺乏内部谓词和丰富的数据类型,因此不能成为一种程序设计语言.自80年代中期,已有许多学者对Datalog进行各种扩充,如增加内部谓词、否定、复杂对象等.[5~10]

在Datalog中增加否定(称为Datalog-Not),使得模型的交不一定是模型,也不一定有唯一的最小模型.[1,2]因此,为了在多个模型中选择一个合理的模型作为最小模型,我们必须给出判断合理模型的标准.其中一种标准就是完美模型.[4]通过比较2个模型的完美关系,我们能选择一个较完美(或最完美)的模型作为最小模型并且作为程序的语义.

文献[1]研究了完美模型的一些简单性质.从理论角度出发,为了揭示完美模型的性质,我们自然要问:(1)完美模型关系是否具有传递性,(2)完美模型关系所构成的有向图具有哪些拓扑结构等等一类问题.为了回答这些问题,本文首次引入完美模型图的概念,通过构造一个Datalog-Not程序P,分析了完美模型关系所构成的有向图具有的拓扑结构.证明了对任一自反有向图G,都能构造一个Datalog-Not程序P,使得P的完美模型图与G同构.从而,在Datalog-Not程序最小模型的研究方向上前进了一步.

1 定 义

若R是形为$L_0:-L_1,...,L_n$的Datalog-Not规则,θ是一个替换满足$L_i\theta$不含变量$(0\leqslant i\leqslant n)$,那么规则$R\theta:L_0\theta:-L_1\theta,...,L_n\theta$称为$R$的一个规则基实例.注意,$L_i\theta$可能是正

* 本文研究得到国家自然科学基金资助.作者黄东斌,1969年生,助工,主要研究领域为逻辑程序设计,知识库与数据库.李磊,1951年生,博士后,教授,主要研究领域为知识库与数据库,逻辑程序设计.

本文通讯联系人:李磊,广州510275,中山大学岭南学院软件研究所

本文1995-12-11收到修改稿

© 中国科学院软件研究所 http://www.jos.org.cn

文字,也可能是负文字.

对任一 Datalog-Not 子句集 S,定义 $S_0=\{S$ 的所有事实$\}\cup\{S$ 的所有规则的所有规则基实例$\}$.

在 S 的 Herbrand 基(记为 B_S)上定义关系"$\rightarrow$"如下:对任意 $K,L\in B_S$,$K\rightarrow L$ 当且仅当存在一规则基实例 $R\in S_0$,满足 K 出现在 R 的体中,L 是 R 的头.

对任意 $K,L\in B_S$,$K>L$ 当且仅当存在一有限关系序列:

$$(K=)N_0\rightarrow N_1,N_1\rightarrow N_2,\ldots,N_{m-1}\rightarrow N_m(=L)$$

满足 $N_j(j=1,\ldots,m)\in B_S$,至少存在 $i(0\leqslant i\leqslant m-1)$使得 N_i 以负文字的形式出现在某规则基实例 $R(\in S_0)$的体,而 N_{i+1}是在 R 的头.

注意,关系"$>$"是传递的,但没有反对称性及自反性.因此,"$>$"不是 B_S 上的半序关系.

设 M_1 和 M_2 是 S 的 Herbrand 模型,称 M_1 比 M_2 更完美(记为 $M_1>^pM_2$),当且仅当对任意 $L\in M_1-M_2$,存在 $K\in M_2-M_1$,使得 $K>L$.注意,对任意 M,$M>^pM$.因此,完美关系 $>^p$ 是自反的.

子句集 S 的 Herbrand 模型 M 是(最)完美的,当且仅当 S 中没有其它模型比 M 更完美.

注意,若 M 是 S 的极小模型,对任意 ΔM,因为 $M-(M\cup\Delta M)=\varnothing$,所以 $M>^p(M\cup\Delta M)$.故在以后的讨论中,我们所指的模型都是指极小模型.

有向图 $G(V,E)$称为子句集 S 的"$>$"关系图,若对任意 $a,b\in V$,$e(a,b)\in E$ 当且仅当 $a>b$.其中 V 是 G 的顶点集,E 是 G 的边集.

自反有向图 $G(V,E)$称为 S 的完美模型图,若 $V=\{M\mid M$ 是 S 的(极小)模型$\}$,且 $e(M_1,M_2)\in E$ 当且仅当 $M_2>^pM_1$.

很明显,Datalog-Not 子句集 S 有完美模型的充要条件是在 S 的完美模型图 $G(V,E)$中,存在点 $v\in V$,使得对任意 $v'(\neq v)\in V$,有 $e(v,v')\notin E$.进一步,S 有唯一完美模型的充要条件是这样的点 v 是唯一的.

2　算　法

引入完美模型图的概念,使我们对子句集 S 的各个模型的完美关系($>^p$)以及完美关系($>^p$)的性质有一个直观的了解.

那么,是否存在一个这样的自反有向图 $G(V,E)$,使得对任意 Datalog-Not 子句集 S,S 的完美模型图与 $G(V,E)$不可同构?

答案是否定的.在下文中,将给出这样的算法,它对任意的自反有向图 $G(V,E)$,构造出一个 Datalog-Not 子句集 S,使得 S 的完美模型图与 $G(V,E)$同构.

第 1 步,对给定的自反有向图 $G(V,E)$,算法 A 构造出一组集合 $M_1,M_2,\ldots,M_n$ 及集合 F,其中 $n=\|V\|$,满足:

1. 若 $e(v_i,v_j)\in E(i\neq j)$,则对任意 $y\in M_j-M_i$,使得存在 $x\in M_i-M_j$ 有 $x>y$.

2. 若 $e(v_i,v_j)\notin E(i\neq j)$,则存在 $y\in M_j-M_i$,使得对任意 $x\in M_i-M_j$ 都没有关系

© 中国科学院软件研究所　http://www.jos.org.cn

$x>y$.

3. 令 $M=M_1\cup M_2\cup\ldots\cup M_n$，对任意 $x,y\in M$，$e(x,y)\in F$ 当且仅当 $x>y$.

据此，可得到"$>$"关系图 $H(M_1\cup M_2\cup\ldots\cup M_n,F)$.

第2步，算法B构造出一组Datalog-Not子句集 S，使得 S 的"$>$"关系图 $H(M_1\cup M_2\cup\ldots\cup M_n,F)$，$S$ 的完美模型图与 $G(V,E)$ 同构.

先定义一些记号：$H_i=\{v_j\mid e(v_j,v_i)\in E\}$

$$I_j=\{v_i\mid e(v_j,v_i)\in E\}$$

$$S_{ij}=S_i\cap S_j-\cup\{S_k\mid k\neq i,k\neq j,1\leqslant k\leqslant n\}$$

$$T_i=S_{ii}$$

单值函数 $f:S\to S$，其中 $S=\cup_{i=1}^{n}S_i$

算法A.

输入：自反有向图 $G(V,E)$，$\|V\|=n$

输出：集合 $M_1,M_2,\ldots,M_n$ 和 F

Step 1.

1.1 初始化集合 T_i 及 $S_{ij}(i,j=1,2,\ldots,n)$ 为空，记录"$>$"关系的集合 F 为空，函数 f 对任意元素无定义.

1.2 在集合 T_i 中增加新点 $IN_i(i=1,2,\ldots,n)$

1.3 在集合 T_i 中增加新点 $OUT_{ij}(i=1,2,\ldots,n)$，若 $v_j\in I_i$

1.4 对所有点 OUT_{ij}，增加边 $e(OUT_{ij},IN_j)$ 到集合 F 中

1.5 令 $f(IN_i)=IN_i$ $\quad i=1,2,\ldots,n$

$f(OUT_{ij})=IN_j$ $\quad i=1,2,\ldots,n;j\in\{j\mid v_j\in I_i\}$

Step 2.

对所有点 OUT_{ij}，做：

对任一 $v_k\in H_i$，做：

2.1 若 $k=j$，则增加新点 P 到 T_j，且标记 P 为 α—型.
增加边 $e(P,OUT_{ij})$ 到集合 F 中，令 $f(P)=IN_j$.

2.2 若 $k\neq j$，则
若存在 $P\in S_{jk}$，满足 $f(P)=IN_j$，则增加边 $e(P,OUT_{ij})$ 到集合 F 中
否则，增加新点 P 到 S_{jk} 中，且标记 P 为 β—型，
并增加边 $e(P,OUT_{ij})$ 到集合 F 中，令 $f(P)=IN_j$.

Step 3.

对所有 α—型的点 P（不妨令 $f(P)=IN_i$），做：
对任一点 $v_j\in H_i$，增加边 $e(OUT_{ij},P)$ 到集合 F 中.

Step 4.

4.1 将 β—型的点重标记为未处理 β—型.

4.2 对所有未处理 β—型点 P（不妨令 $f(P)=IN_i,P\in S_{ij}$）做：

4.2.1 对任一点 $v_k\in H_i$ 且 $k\neq j$，增加边 $e(OUT_{ki},P)$ 到 F 中

4.2.2 对任一点 $v_k\in H_j$ 且 $k\neq i$，做：
若边 $e(P,OUT_{ki})\in F$，则增加边 $e(OUT_{ki},P)$ 到 F 中. 否则，
若 S_{ik} 非空，且存在 $Q\in S_{ik}$ 满足 $f(Q)=IN_i$
则增加边 $e(Q,P)$ 到 F 中. 否则，增加新点 Q 到 S_{ik} 中且标记 Q 为 β—型，
并增加边 $e(Q,P)$ 到 F 中，令 $f(Q)=IN_i$.

4.3 将所有未处理 β—型点重标记为已处理 β—型点.

4.4 若存在 β—型点，则继续 Step 4.

Step 5.

令 $M_i=T_i\cup S_{ij}\cup\{modle\}$ $i,j=1,2,\ldots,n$

定理1. 在 $G(V,E)$ 中，若 $e(V_j,V_i)\in E$，则 $M_i>^{p}M_j(i\neq j)$.

定理2. 在 $G(V,E)$ 中，若 $e(V_j,V_i)\notin E$，则 $G_i\cap(M_j-M_i)=\varnothing$，其中 $G_i=\{x\mid x\in M_1\cup M_2\cup\ldots\cup M_n,f(x)=IN_i\}$.

定理3. 若 $e(p_1,p_2)\in F$，则 $f(p_1)=f(p_2)$.

© 中国科学院软件研究所 http://www.jos.org.cn

定理 4. 在 $G(V,E)$ 中，若 $(V_j,V_i)\notin E(i\neq j)$，则 $M_i>^p M_j$ 不成立.

定理 5. 在 $G(V,E)$ 中，$(V_j,V_i)\in E$，当且仅当 $M_i>^p M_j$.

（上述定理的证明见附录）

算法 *B*.

输入：算法 A 得到的模型 $M_1,M_2,\ldots,M_n$ 及集合 F

输出：一组 Datalog-Not 子句集 S

Step 1　构造以下子句集：

evil1：— ¬evil2，trigger.

evil2：— ¬evil3，trigger.

evil3：— ¬evil1，trigger.

trigger：— ¬modle.

trigger：— lock.

Step 2　对所有 $M_i(i=1,2,\ldots,n)$ 做：

2.1　不妨设 $M_i=\{r_1,r_2,\ldots,r_p,\text{modle}\}$，增加以下子句：

modle：$-r_1,r_2,\ldots,r_p$.

2.2　对任意 $x\in M(=\bigcup M_i)$ 且 $x\notin M_i$，增加子句：

trigger：$-r_1,r_2,\ldots,r_p,x$.

Step 3　对任意 $e(x,y)\in F$，增加子句：

y：$-\neg x$，lock.

定理 6. 算法 B 构造的子句集 S，其所有极小 Herbrand 模型是 $M_1,M_2,\ldots,M_n$（证明见附录）.

定理 7. 对自反有向图 $G(V,E)$，存在一个 Datalog-Not 子句集 S，S 的完美模型图与 $G(V,E)$ 同构.

证明：由算法 A，B 可知，存在一个 Datalog-Not 子句集 S，S 的所有极小 Herbrand 模型是 $M_1,M_2,\ldots,M_n$（定理 6）.

设子句集 S 的完美模型图是 $H(W,F)$，其中 $W=\{M_1,M_2,\ldots,M_n\}$.

令图 G 和图 H 的顶点影射函数 f 如下：

$f:V\to W$，满足 $f(v_i)=M_i\quad(i=1,2,\ldots,n)$

由定理 5 知 $e(v_j,v_i)\in E$ 当且仅当 $M_i>^p M_j$，即 $f(v_i)>^p f(v_j)$.

另一方面，由完美模型图定义，$e(M_j,M_i)\in F$，当且仅当 $M_i>^p M_j$.

故 $e(v_j,v_i)\in E$，当且仅当 $e(f(v_j),f(v_i))\in F$.

所以 $H(W,F)$ 与 $G(V,E)$ 同构. □

定理 8. 自反有向图 $G(V,E)$，存在 $v\in V$，对任意 $v'(\neq v)\in V$，$e(v,v')\in E$ 都不成立，当且仅当存在一个 Datalog-Not 子句集 S，S 有完美模型且 S 的完美模型图与 $G(V,E)$ 同构.

证明：（充分性）由定义可得.

（必要性）由算法 A，B 可知，存在一个 Datalog-Not 子句集 S，S 的完美模型图与 $G(V,E)$ 同构.

设 v 与模型 M 对应，v' 与模型 M' 对应.

因为 $e(v,v')\in E$，当且仅当 $M'>^p M$.

而这样的 v' 是不存在的，故 M' 不存在.

因此 M 是完美模型. □

© 中国科学院软件研究所　http://www.jos.org.cn

3 例 子

在此，将构造一个 Datalog-Not 子句集 S，它有 3 个模型 M_1, M_2 和 M_3，满足 $M_3 >^p M_2$，$M_2 >^p M_1$.

按算法 A，得到“>”关系图如图 1 所示.

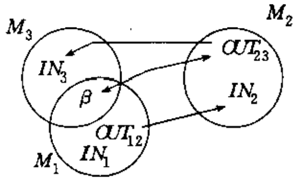

图1

$M_1 = \{IN_1, OUT_{12}, \beta, \text{modle}\}$

$M_2 = \{IN_2, OUT_{23}, \text{modle}\}$

$M_3 = \{IN_3, \beta, \text{modle}\}$

按算法 B，能得到子句集 S 如下（注意：S 有 3 个最小模型 M_1, M_2 和 M_3）.

evil1 :- ¬evil2, trigger.
evil2 :- ¬evil3, trigger.
evil3 :- ¬evil1, trigger.
trigger :- ¬modle.
trigger :- lock.

modle :- IN_1, OUT_{12}, β.
modle :- IN_2, OUT_{23}.
modle :- IN_3, β.

trigger :- $IN_1, OUT_{12}, \beta, IN_2$.
trigger :- $IN_1, OUT_{12}, \beta, OUT_{23}$.
trigger :- $IN_1, OUT_{12}, \beta, IN_3$.
trigger :- IN_2, OUT_{23}, IN_1.
trigger :- IN_2, OUT_{23}, OUT_{12}.
trigger :- IN_2, OUT_{23}, β.
trigger :- IN_2, OUT_{23}, IN_3.
trigger :- IN_3, β, IN_1.
trigger :- IN_3, β, OUT_{12}.
trigger :- IN_3, β, IN_2.
trigger :- IN_3, β, OUT_{23}.

IN_2 :- $\neg OUT_{12}$, lock.
IN_3 :- $\neg OUT_{23}$, lock.
β :- $\neg OUT_{23}$, lock.
OUT_{23} :- $\neg\beta$, lock.

从而，我们可以看出，关系“>”有传递性，然而 $>^p$ 却没有传递性.

由于 β :- $\neg OUT_{23}$, lock. 及 OUT_{23} :- $\neg\beta$, lock. 因此子句集 S 不可分层. 从而得到：

推论. 存在一个不可分层的子句集 S，S 有唯一完美模型.

4 小 结

本文分析了 Datalog-Not 子句集模型间的完美关系的结构. 完美关系并不具有传递性，而它定义的基础“>”关系是传递的，给构造算法带来本质的困难. 从本文的结果可以得出，“>”关系的传递性并不影响模型的完美关系的传递性.

参考文献

1 Ceri S, Gottlob G, Tanca L. Logic programming and database. Berlin Heidelberg: Springer-Verlag, 1990. 215

© 中国科学院软件研究所 http://www.jos.org.cn

～225.

2 Lloyd J W. Foundation of logic programming. Berlin Heidelberg: Springer-Verlag, 1984.

3 Chandra A, Harel D. Horn clause queries and generalizations. Journal of Logic Programming, 1985, **1**:1～15.

4 Ullman J D. 数据库与知识库系统原理(卷Ⅱ新技术). 徐秋元等译. 西安:陕西省新闻出版局, 1991.

5 Lifschitz V. On the declarative semantics of logic programs with negation. Proc. of Workshop on foundations of deductive databases and logic programming, August 1986. 420～432.

6 Tsur S, Zanilol C. LDL: a logic-based data-language. Proc. 12th Int Conf. on Very Large Databases, Kyoto, Japan, 1986.

7 Beeri C, Naqri S. Sets and negation in a logic database language (LDL1). ACM Conf. on Principles of Database Systems, San Diego, 1987. 21～37.

8 Chen Weidong. HILOG: a foundation for higher-order logic programming. J. Logic Programming, 1993, **15**:187～230.

9 Serge Abiteboul. COL: a logic-based language for complex objects. Workshop on Database Programming Language, 1987.

10 Gabriel M Kuper. Logic programming with sets. ACM SIGACT/SIGMOD Symposium on Principle of Database Systems, 1987.

附录

定理 1. 在 $G(V,E)$ 中,若 $e(V_j,V_i)\in E$,则 $M_i>^pM_j(i\neq j)$.

证明:只需证:对任意 $x\in M_i-M_j$,存在 $y\in M_j-M_i$,使 $y>x$,即 $(y,x)\in F$.

考察任意 $x\in M_i-M_j$,有以下 4 种情况:

(1) $x=IN_i$,则存在 $OUT_{ji}\in M_j-M_i$,使 $(OUT_{ji},IN_i)\in F$.

(2) $x=OUT_{ir}$,因 $v_j\in H_i$,由 Step2 知存在 $P\in T_j$ 或 $P\in S_{rj}$,即 $P\in M_j-M_i$,使 $(P,x)\in F$.

(3) x 为 α 型点且 $x\in T_i$,则 $f(x)=IN_i$.

因 $v_j\in H_i$,则由 Step3 有 $(OUT_{ji},x)\in F$,而 $OUT_{ji}\in M_j-M_i$.

(4) x 为已考虑 β 型点,不妨设 $x\in S_{ik}$　$(k\neq i,k\neq j)$.

A)若 $f(x)=IN_i$,由 Step4.2.1 有 $OUT_{ji}\in T_j$,使 $(OUT_{ji},x)\in F$

B)若 $f(x)=IN_k$,因 $v_j\in H_i$,由 Step4.2.2 知,存在 $y=OUT_{jk}$ 或 $y\in S_{jk}$,即有 $y\in M_j-M_i$,使 $(y,x)\in F$.

故总存在 $y\in M_j-M_i$,使 $(y,x)\in F$,即 $y>x$,从而 $M_i>^pM_j$. □

定理 2. 在 $G(V,E)$ 中,若 $e(V_j,V_i)\notin E$,则 $G_i\cap(M_j-M_i)=\varnothing$,其中

$$G_i=\{x\mid x\in M_1\cup M_2\cup\ldots\cup M_n,f(x)=IN_i\}.$$

证明:对任意 $x\in G_i$,若

(1)在 Step1 增加,则 $x=IN_i$ 或 $OUT_{ki}(k\neq j$ 否则,$(v_j,v_i)\in E)$,因为 $OUT_{ki}\in T_k$,故 $x\notin M_j$.

(2)在 Step2 增加

A)在 Step2.1 增加,则 $f(x)=IN_i$,$x\in T_i$,故 $x\notin M_j$.

B)在 Step2.2 增加,则 $f(x)=IN_i$,$x\in S_{ik}(i,j,k$ 各不相同),故 $x\notin M_j$.

(3)在 Step4 增加,因 $f(x)=IN_i$,$x\in S_{ik}\subset M_i(k\neq i)$,故 $x\in M_i$,所以 $x\notin M_j-M_i$,从而 $G_i\cap(M_j-M_i)=\varnothing$. □

定理 3. 若 $e(p_1,p_2)\in F$,则 $f(p_1)=f(p_2)$.

证明:因为在算法 A 各步中,新增的关系 $(p_1,p_2)\in F$,必满足 $f(p_1)=f(p_2)$. □

定理 4. 在 $G(V,E)$ 中,若 $(V_j,V_i)\notin E(i\neq j)$,则 $M_i>^pM_j$ 不成立.

证明:要证 $M_i>^pM_j$ 不成立,仅需证存在 $x\in M_i-M_j$,对任意 $y\in M_j-M_i$,使 $y>x$ 不成立,即 $(y,x)\notin F^*$(是 F^* 的传递闭包).

考察 $x=IN_i$,假设存在 $y\in M_j-M_i$,使 $(y,IN_i)\in F^*$,则存在序列 $t_0(=y),t_1,\ldots,t_m(=IN_i)$,使 $(t_0,t_1),(t_1,t_2),\ldots,(t_{m-1},t_m)\in F(m\geqslant 1)$.

由定理 3 知 $f(t_k)=f(t_m)$,$k=0,1,\ldots,m$.

© 中国科学院软件研究所　http://www.jos.org.cn

而 $f(t_m)=IN_i$，故 $t_k\in G_i$.

由定理 2，$t_k\notin M_j-M_i$，$k=0,1,\ldots,m$.

但与 $y=t_0\in M_j-M_i$ 矛盾.

故存在 x，$x=IN_i$，对任意 $y\in M_j-M_i$，使 $(y,x)\notin F^*$.

因此，$M_i>^pM_j$ 不成立. □

定理 5. 在 $G(V,E)$ 中，$(V_j,V_i)\in E$，当且仅当 $M_i>^pM_j$.

证明：由定理 1 及定理 4 可得. □

定理 6. 算法 B 构造的子句集 S，其所有极小 Herbrand 模型是 $M_1,M_2,\ldots,M_n$.

证明：对子句集 S 任一最小模型 M，有

trigger$\notin M$，否则由 Step1 中的子句，将导致 evil1$\in M$ 且 evil1$\notin M$.

从而 lock$\notin M$，modle$\in M$

由算法 B，容易验证 $M_1,M_2,\ldots,M_n$ 是子句集 S 的最小模型.

假设 M' 是 S 另一最小模型（$M'\neq M\quad i=1,2,\ldots,n$），则 modle$\in M'$，由 Step2.1 知存在 M_i 使 $M'\supset M_i$，因此 M' 不是最小模型，矛盾.

所以，$M_1,M_2,\ldots,M_n$ 是 S 所有最小 Herbrand 模型. □

THE NECESSARY AND SUFFICIENT CONDITION OF PERFECT MODEL

Huang Dongbin

(*The Science & Technology Department of the Telecommunication Bureau of Guangzhou Guangzhou* 510055)

Li Lei

(*Institute of Software of Lingnan College Zhongshan University Guangzhou* 510275)

Abstract In Datalog-Not clauses, the existence of the unique minimal model is no longer satisfied while it is satisfied in Datalog clauses. The authors need a policy to select a model as the semantics of a Datalog-Not program P among all of the models of the program P. The perfect model is a preferable model. Unfortunately, the necessary and sufficient condition of the perfect model has not been considered since the perfect relationship is too complex. In this paper, the perfect model is deeply analyzed and its necessary and sufficient condition is given. Furthermore, the structured algorithm is carried out and its correctness is proven to reach the following theorem: for any reflective directive graph G, the authors can construct the Datalog-Not clauses P whose perfect-model graph has the same structure with G.

Key words Logic program, Herbrand model, stratified program, perfect model.

© 中国科学院软件研究所 http://www.jos.org.cn