计算机科学 1994 Vol. 21 №. 2

逻辑程序的语义问题（Ⅱ）

王怀民
（国防科技大学　长沙 410073）

下面几节讨论模型论途径的说明语义，可以克服 Clark 语义的上述缺点。

4. 最小模型语义

最小模型语义仅适合于正逻辑程序。我们非常熟悉的基于 Horn 逻辑的 Prolog 程序采用这种语义规则。为了更好地理解逻辑程序的模型语义的新发展，这里首先考虑最小模型语义。本节仅涉及 2-值解释。

例 4.1 设程序 P 为

able-mathematician (x) ←physicist (x);

physicist (einstein);

businessman (iacocca)。

P 有许多 Herbrand 模型。其中最大的 Herbrand 模型描述了这样的可能世界：Einstein 和 Iacocca 同时是商人、物理学家和优秀的数学家。正常情况下，我们不能接受该模型作为 P 的语义模型。实际上 P 不能导出 "physicist (iacocca)" 和 "businessman (einstein)"。这些事实往往不成立。

P 有唯一的最小模型 M_P= {physicist (einstein), businessman (iacocca), able-mathematician (einstein)}。这个模型看上去正确地反映了 P 的意义。同时也反映了封闭世界假设的传统形式：如果不存在某个正断言为真的理由，那么认为该断言为假。

定理 4.1（Van Emden 和 Kowalski, 1976）　每个正逻辑程序有唯一的最小 Herbrand 模型 M_P。

由此可以给出最小模型语义的定义。

定义 4.1（最小模型语义）　一个正逻辑程序 P 的最小模型语义由 P 的最小 Herbrand 模型 M_P 决定。

最小模型语义直观地反映了正逻辑程序的意义。其中的基本思想是：尽可能极小化正信息，使其限制在由 P 显式蕴含的事实，而其它信息均为假。换句话说，最小模型语义基于封闭世界假设。

最小模型语义可以克服 Clark 语义的一些缺点。例如上节给出的程序 P_1 和 P'_1 的最小 Herbrand 模型是一致的，即为：{natural-number (0), natural-number (succ (0)), natural-number (succ (succ (0))), …}。这正是所希望的。类似地，上节程序 P_3 的最小 Herbrand 模型 M_{P3}= {edge (a, b), edge (c, d), edge (d, c), reachable (a), reachable (b)} 也是我们实际希望的。

最小模型语义的另一个优点是具有自然的不动点性质。

定义 4.2（Van Emden-Kowalski 操作）　设 P 是一个正逻辑程序，$I \in \mathcal{I}$ 是 P 的一个解释，A 是一个基原子，$\Psi(I)$ 是如下定义的解释：

(1) $\Psi(A)=1$，如果 P 中存在子句 $A \leftarrow A_1, \cdots, A_n$ 使得 $I(A_i)=1$，$i=1, \cdots, n$；

(2) $\Psi(A)=0$，其它。

定理 4.2（Van Emden 和 Kowalski, 1976）　Ψ 有最小不动点，且为 M_P。进一步，M_P 可由 Ψ 迭代 ω 次获得（ω 表示第一个无穷极限序数）。即 Ψ 的迭代序列 $\{\Psi^{\uparrow n}\}$ 是单调增的，且有一个不动点 $\Psi^{\uparrow\omega}=M_P$。

最小模型语义比 Clark 语义强。

定理 4.3　设 P 是一个正逻辑程序，p 是一个闭公式。comp (P) $\vdash$ p，则 $M_P \vdash$ p。

最小模型语义的一个严重缺点是仅适合于正逻辑程序。逻辑程序在一般情况下没有最小 Herbrand 模型。例如程序 $p \leftarrow\!\!\rightarrow q$ 有两个极小模型 $\{\{p\}, \phi\}$ 和 $\{\{q\}, \phi\}$，但没有最小模型。下节介绍的理想模型语义把最小模型语义的思想推广到更广的一类逻辑程序之上。

5. 理想模型语义

否定在实际应用中十分重要，含有否定的程序子句显然增强了逻辑程序的表达能力。但寻找这类程序的适当语义较之正逻辑程序要困难得多。这一节介绍 T. Przymusinski 提出的理想模型语义（Perfect Model Semantics）（1988 年），它将最小模型语义的思想推广到一类可含有负文字前提的逻辑程序上。这一节的大部分仅涉及 2-值解释（模型）。为了说明理想模型的基本思想，我们首先考虑一个例子。

例 5.1 假设我们知道物理学家是有才干的数学家；而通常情况下，商人在工作中总是回避（高明的）数学家，除非这位商人碰巧有很强的数学基础。另外，我们还知道 Iacocca 是一个商人，Einstein 是物理学家。我们可以用下面的逻辑程序 P 表达这些事实：

avoids-math (x) ←businessman (x), →able-mathematician (x); (1)

able-mathematician (x) ←physicist (x); (2)

businessman (iacocca); (3)

physicist (einstein)。 (4)

这个程序没有最小模型，只有两个极小模型：

M_1 = {businessman (iacocca), physicist (einstein), able-mathematician(einstein), avoids-math(iacocca)}

M_2 = {businessman (iacocca), physicist (einstein), able-mathematician (einstein), able-mathematician (iacocca)}

M_1 给出的是 Iacocca 回避数学家的可能世界，当然这时 Iacocca 不是有才干的数学家。另一方面，M_2 给出的是 Iacocca 为有才干的数学家的可能世界。

通常逻辑程序的意义包含了某种形式的封闭世界假设，即以某种方式极小化正信息。基于这种认识，考虑 P 的极小模型作为其语义模型是自然的。如果考虑 P 的极小模型的集合 MOD (P) 作为 P 的语义，那么除 businessman (iacocca) 这个在 P 中显式给出的正信息外，我们不能从 MOD (P) 中得到更多的关于 Iacocca 的正信息。但我们总觉得 P 还包含了更多的关于 Iacocca 的正信息。简言之，MOD (P) 作为 P 的语义并不十分贴切。下面的问题是取 P 的哪一个极小模型更合理。

按照我们对程序 P 的背景的陈述，我们更倾向于用 M_1 作为 P 的语义。容易说明其中取 M_1 舍 M_2 的原因：P 作为程序反映出的某种过程性质。P 中的子句（1）在逻辑上等价如下子句：

able-mathematician (x) ∨ avoids-math (x) ←businessman (x)。 (5)

设 P′ 是用子句（5）代换 P 中子句（1）所得到的理论。显然 M_1，M_2 是 P′ 的极小模型。子句（1）和子句（5）在表达知识的意图上存在差别：后者没有考虑 able-mathematician (x) 和 avoids-math (x) 成立与否的优先次序，即认为这两个性质的出现有同等的可能。因此应该同时采用 M_1 和 M_2 作为反映 P′意图的语义。而另一方面，直观地看，子句（1）赋予了谓词 able-mathematician 和 avoids-math 在极小化过程中的优先差别：具体地讲，应首先考虑谓词 able-methematician 所陈述性质的真假，如果没有绝对的理由证实其为真，则应认为其为假。我们可以说子句（1）赋予 able-mathematician 在极小化（即无根据证明其为真，则取其为假的过程）方面优先于谓词 avoids-math。下面的子句反映了一个与上面的优先级相反的情况：

able-mathematician (e) ←physicist (e), →avoids-math (e)。

该子句是说：如果 e 是物理学家，且没有特别迹象表明 e 回避数学，那么我们可以假定 e 是一位有才干的数学家。这里谓词 avoids-math 在极小化方面有高于谓词 able-mathematician 的优先级。也就是说，应首先考虑 avoids-math (e) 为假，除非有特别的理由证明其为真，然后再考虑对 able-mathematician (e) 的极小化。

可以看出，对于子句 B←A（这里 A，B 是两个原子），如果极小化 B（即取 B 为假），那么同时 A 也立刻被极小化，因为对于满足 B←A 的（极小）模型，B 为假，则 A 必为假。这就是说，A 总是在 B 之前或同时被极小化。

根据上面的讨论，我们可以得出这样的

结论，程序子句的语法决定了基原子在极小化方面的相对次序。其规则如下：

（1）子句的负前提具有高于子句头的优先级；

（2）子句的正前提的优先级不低于子句头。

为了形式地描述规则（1）和（2），这里引入逻辑程序P的相关图G_P的概念(这里假设P已经实例化)。G_P的结点是$\mathscr{B}$中的基原子，设A，B$\in\mathscr{B}$。在G_P中从B到A有一条直接边当且仅当P中有一个子句以A为头，且其前提中含有B或¬B，对于后者该直接边称为负的。

定义 5.1（优先关系） 设A，B$\in\mathscr{B}$，B的优先级高于A（记为A<B），如果G_P中存在从B到A的路径经过至少一条负边。称上面定义的基原子之间的关系<为基原子之间的优先关系。如果从B到A有一条路径，则记A≤B。类似地，我们可以用谓词符号代替基原子，定义谓词符号之间的优先关系。

理想模型是在优先关系的基础上定义的，其基本思想是在极小模型中取这样一个模型使得优先级高的原子在该模型中首先被极小化（即赋其真值为假）。精确一点说，假设M是P的模型，N是通过往M中加一些原子再去掉一些原子得到的新模型。我们说N较之M更理想当且仅且被加入的原子（即在N中赋予真）的优先级低于被删去的原子（即在N中赋予假）。由优先关系的定义，被加入的原子是由于被删去的原子被极小化（即赋予假）而被确证的。P的一个模型M是理想的，如果P不存在较之M更理想的模型。

定义 5.2（理想模型，Przymusinski，1988） 设M，N是逻辑程序P的两个互不相同的模型。称N较之M更理想（简记为N≪M），如果$\forall$ A$\in$N−M，存在一个优先级高于A的原子B，使得B$\in$M−N。我们说P的模型M是理想的，是指不存在P的另一个模型N使得N≪M。称关系≪是模型之间的理想关系。

定理 5.1（Przymusinski，1988） 理想模型是极小模型。对于正逻辑程序，最小模型与理想模型是一致的。

例 5.2 对于例5.1中的程序，M_1是理想的。因为able-mathematician>avoids-math，所以$M_1\ll M_2$，但$M_2\ll M_1$不成立。

不幸的是并非每一个逻辑程序都有理想模型。

例 5.3 程序{q←¬p，p←¬q}有两个极小模型M_1={p}和M_2={q}。因为p<q且q<p，则$M_1\ll M_2$且$M_2\ll M_1$。所以该程序没有理想模型。

不难分析产生这种现象的原因：理想模型的概念基于基原子上的优先关系，而优先关系可能产生冲突，即在程序的相关图G_P中出现环路，由此导致理想模型不存在。为了消除否定的循环定义，Apt，Blair，Walker和Van Gelder等人提出了分层的逻辑程序，后来Przymusinski又将这一概念推广为局部分层的逻辑程序。

定义 5.3 一个逻辑程序P是分层的（或局部分层的），如果可以在所有谓词符号集S（或Herbrand基$\mathscr{B}$）上得到一个划分S_1，S_2，…，S_α，…，$\alpha<\lambda$使得对于P中任一子句（或子句的实例）：

$$C\leftarrow A_1,\ \cdots,\ A_m,\ \neg B_1,\ \cdots,\ \neg B_n$$

其中A_i，B_j，C（i=1，…，m，j=1，…，n）是原子，满足：

(1) $\forall$ i. stratum (A_i) ≤stratum (C)；

(2) $\forall$ j. stratum (B_j) <stratum (C)。

这里stratum (A) =α，如果A的谓词符号属于α层S_α（或A属于S_α）。任一满足上述条件的划分称为P的一个分层（或局部分层）。

根据上述定义，程序P的分层决定了原子之间的优先级。这里较低的层次对应较高的优先级。例如，例5.1的程序是分层的，其中的一个分层为S_1={able-mathematician}，S_2={businessman，physicist，avoids-math}。

命题 5.1 每一个分层程序也是局部分层的。

命题 5.2 一个逻辑程序是分层的当且仅当它的谓词优先关系$<_P$是一个偏序。一

个逻辑程序是局部分层的，如果它的基原子优先关系$<$是一个偏序，且基原子关于$<$的每一个增长序列是有限的。

下面是一个局部分层但不是分层程序的例子。

例 5.4 下面的程序定义了偶数：

even (0)；

even (s (x)) ←¬even (x)．

这里 s (x) 是后继函数。这个程序不是分层的，因为谓词 even 涉及到关于其自身的负递归，即 even$<_P$ even。但是，P 是局部分层的。因为不难看出基原子之间的优先关系$<$是偏序，而且基原子的每一个递增序列形如：

even (s (s (s (…)))) $<$even (s (s (…))) $<$even (s (…)) $<$…$<$even (s (0)) $<$even (0)，是有限的。

定理 5.2（Przymusinski，1988） 每个局部分层的程序 P 有唯一的理想模型 M_P，而且 M_P 较之 P 的其它任何模型更理想，即对于 P 的其它模型 M，$M_P \ll M$。

由此可以定义局部分层程序的理想模型语义。

定义 5.4（理想模型语义） 设 P 是局部分层的程序。P 的理想模型语义由 P 的唯一理想模型 M_P 决定。

当然理想模型语义不仅仅限于局部分层的程序。由定理 5.1，正逻辑程序的理想模型语义等价于最小模型语义。可见理想模型语义是最小模型语义的扩充。下面的结论指出理想模型语义强于 Clark 语义。

定理 5.3 如果 P 是局部分层的程序且 comp (P) 是一致的。那么被 comp (P) 蕴含的闭公式也被理想模型语义蕴含。

理想模型语义消除了第 3 节中 Clark 语义的某些不够自然的方面。例如，对于例 3.3 给出的程序 P_4，其理想模型为 {bird (tweety)，fly (tweety)}，它反映了 P_4 的意图。

这里局部分层程序的理想模型也具有迭代最小不动点和迭代最小模型的性质。

定义 5.5（Van Emden-Kowalski 操作的推广） 设 P 是某一逻辑程序，J 是任意的 3-值解释。$I \in \mathcal{I}$ 是 2-值解释，A 是一个基原子。Ψ_J (I) 是如下定义的 2-值解释：

(1) Ψ_J (I) (A) $=1$，如果 P 中存在一子句 $A \leftarrow L_1, \cdots, L_n$ 使得 $\forall$ i，或者 $\hat{J}(L_i) = 1$，或者 L_i 是一个原子且 $I(L_i) = 1$；

(2) Ψ_J (I) $=0$，其它

直观地讲，J 表示当前知其为真或假的事实。Ψ_J (I) 包含所有由程序 P 在如下假设下一步推出的原子事实。这些假定是所有在 J 中成立的（肯定的和否定的）事实，以及所有在 I 中为真的正事实。

不难看出上一节定义的 Van Emden-Kowalski 操作 Ψ 是 Ψ_J 的特例，其中 $J = \langle \phi; \phi \rangle$。

命题 5.3 操作 Ψ_J 是单调的，而且有最小不动点 Ψ_J^{ω}。

直观地讲，最小不动点 Ψ_J^{ω} 包括了在知道 J 的情况下由 P 推导出的所有正原子事实。设 $\{S_1, S_2, \cdots, S_\alpha, \cdots\}$，$\alpha < \lambda$，是一个局部分层程序 P 的局部分层，对于 $\beta \leqslant \lambda$，设：

$\mathcal{H}_\beta = \bigcup_{\alpha<\beta} S_\alpha$，

显然 $\mathcal{H} = \mathcal{H}_\lambda$

我们构造如下（3-值）解释的（超穷）序列 $\{I_\alpha : \alpha \leqslant \lambda\}$：

$I_0 = \{\phi; \phi\}$；

$I_{\alpha+1} = \{\Psi_{I_\alpha}^{\omega}; \mathcal{H}_{\alpha+1} - \Psi_{I_\alpha}^{\omega}\}$；

$I_\delta = \sum_F \{I_\alpha \mid \alpha < \delta\}$ δ 是极限。

直观地讲，迭代 $I_{\alpha+1}$ 是由 I_α 这样得到的：

(1) 取操作 Ψ_{I_α} 的最小不动点作为真原子事实集，即在知道 I_α 的情况下由 P 导出的原子；

(2) 取 $\Psi_{I_\alpha}^{\omega}$ 在 $\mathcal{H}_{\alpha+1}$ 中的补集作为假原子的集合，即在第 $\alpha+1$ 层不能被导出的事实为假。

可以证明序列 $\{I_\alpha\}$ 是 F-递增，最终到达解释 I_λ。

定理 5.4 局部分层的逻辑程序的理想模型 $M_P = I_\lambda$，而且 M_P 是 Ψ_{M_P} 的最小不动点。

虽然局部分层程序的理想模型有许多良好的性质，但它仅适合于这一类程序，有许多有意义的逻辑程序不是局部分层的。

例 5.5 考虑如下程序 P：

p (1, 2) ←；q (X) ←p (X, Y), ¬q (Y)。

实例化后，P 为如下形式：

p (1, 2) ←； (6)

q (1) ←p (1, 2), ¬q (2)； (7)

q (1) ←p (1, 1), ¬q (1)； (8)

q (2) ←p (2, 2), ¬q (2)； (9)

q (2) ←p (2, 1), ¬q (1)。 (10)

这个程序不是局部分层的，因为 P 的优先关系不是偏序：q (1) <q (2) 且 q (2) <q (1)。但 P 的意义是明确的，可以用 2-值模型 M= {p (1, 2), q (1)} 指称。并且通常的 Prolog 系统可以推出与 M 一致的结果。实际上 P 在语义上等价于由子句 (6) 和 (7) 组成的局部分层程序 P'。子句 (8)、(9)、(10) 看上去与 P 的意义毫不相干，但正是这几个子句破坏了 P 的局部分层性质。

有三种途径扩充理想模型语义：弱理想模型语义，稳定模型语义和良基语义。与局部分层程序的理想模型思想相似，弱理想模型语义基于程序的划分分层。其特别之处在于将对程序的静态划分变为对程序的动态划分。这里不进一步讨论弱理想模型语义，我们将重点介绍稳定模型语义和良基语义。

6. 稳定模型语义

稳定模型 (Stable Model) 由 Gelfond 和 Lifschitz 提出 (1988)。这种模型采用逻辑程序上一个自然变换的不动点定义。我们首先给出作用于 2-值解释上的 Gelfond-Lifschitz 操作 Γ。

定义 6.1 (Gelfond-Lifschitz 操作) 设 P 是一个逻辑程序，I 是 P 的解释。P 模 I 的 GL-变换是指用以下两步化简规则作用于 P 得到的新程序 P/I：

(1) 从 P 中去掉所有这样的子句，其中含有一个否定前提 L=¬A 使 Î (L) =0。由此得到程序 P'；

(2) 如果 P' 的子句前提中有文字 L=¬A 满足 Î (L) =1，则将该文字从该子句的前提中去掉，得到新的子句。由此得到程序 P/I。

由于 P/I 是正逻辑程序，由定理 4.1，P/I 有唯一的最小模型 J。我们定义 Γ_P (I) =J。

直观地讲，Γ_P (I) 意指在知道 I 的情况下由 P 可以推出的原子事实。因为这时规则 (1) 删去的子句在推理中不能使用，规则 (2) 去掉的前提已由 I 确认成立，因此 Γ_P (I) 实际是由 P/I 可以推出的结论。

命题 6.1 (Gelfond & Lifschitz, 1988) 设 P 是逻辑程序。Γ_P 的不动点是 P 的极小模型。

定义 6.2 (稳定模型语义 Gelfond & Lifschitz, 1988) 逻辑程序 P 的一个 2-值解释 I 被称为稳定模型，如果 Γ_P (I) =I。如果 P 有唯一的稳定模型 M_P，则其稳定模型语义由 M_P 决定。

例 6.1 考虑例 5.5 中的程序。取 M= {p (1, 2), q (1)}，不难得到 P/M 为：

p (1, 2) ←； (11)

q (1) ←p (1, 2)； (12)

q (2) ←p (2, 2)。 (13)

显然 P/M 的最小模型为 M，即 Γ_P (M) =M。因为 M 是 P 的一个稳定模型。可以验证 M 是 P 的唯一稳定模型。因此 M 是 P 的稳定模型语义。

定理 6.1 (Gelfond & Lifschitz, 1988) 每一个局部分层的逻辑程序 P 有唯一的稳定模型，这就是 P 的理想模型。

例 6.1 和定理 6.1 表明稳定模型是理想模型的扩充。我们可以进一步把 2-值稳定模型扩充到 3-值上。首先在 $\mathscr{L}$ 中增加命题符号 u 表示“不知道”或“未定义”的状态，并假设任意解释满足 I (u) =1/2，Î (¬u) =1/2。因此总是可以用 u 代换¬u。以下结论是对定理 4.1 的 3-值推广。

定理 6.2 每个非负的逻辑程序 P 有唯一的最小 3-值模型 (在标准序的意义下)。

下面将 Gelfond-Lifschitz 操作 Γ 扩充为 3-值操作 Γ^*。

定义 6.3 设 P 是逻辑程序，I 是 P 的 3-值解释，P 的模 I 的扩充 GL-变换是由 P 作用如下三步简化规则得到的新程序 P/I：

(1) 从 P 中去掉所有这样的子句，其中含有一个否定前提 L=¬A 使得 Î (L) =0。由

此得到程序 P′；

（2）如果 P′ 的子句前提中有文字 L＝—A 使得 $\hat{I}(L)=1/2$，则将该文字用 u 代替，由此得到程序 P″；

（3）如果 P″的子句前提中有文字 L＝—A 使得 $\hat{I}(L)=1$，则将该文字从该子句的前提中去掉，得到新的子句。由此得到 P/I。

显然 P/I 是非负程序。由定理 6.2，P/I 有唯一的最小 3-值模型 J。我们定义 $\Gamma_P^*(I)=J$。Γ_P^* 的不动点定义为 P 的 3-值稳定模型。

定义 6.4 一个逻辑程序 P 的 3-值解释 I 称为 3-值稳定模型，如果 $\Gamma_P^*(I)=I$。P 的 3-值稳定模型语义由 P 的 3-值稳定模型集合 MOD（P）确定。

下一节我们将看到每一个程序至少有一个 3-值稳定模型。

例 6.2 设 P 为：c←→d；a←→b；b←→a. 取 M＝（{c}；{d}），则 P/M 为：c←；a←u；b←u. 不难看出 $\Gamma_P^*(M)=M$。因此 M 是 P 的 3-值稳定模型。

稳定模型的主要缺点是没有给出一般的类似最小不动点迭代生成的良好构造方法。另外稳定模型语义有时也会与我们希望的语义相悖。下面我们分析一个 2-值稳定模型的例子。

例 6.3 我们首先陈述一个日常生活中的会话场景：妻子对丈夫命令式地说道："明天我们上街去！""好！明天我们上街"丈夫无可奈何地答道；接着又模棱两可地补充道："明天不下雨我们就上街。依我看，明天不是刮风就是下雨。"妻子不解地问："明天我们到底上不上街？"显然妻子从丈夫是是而非的回答中没有得到确切的信息。但如果我们将丈夫的回答表示成如下程序 P：

p←→p（表示明天上街）；

p←→a（表示如果不下雨，明天上街）；

b←→a；

a←→b（表示不是刮风就是下雨）。

则 P 有唯一的稳定模型 M＝{p，b}，即 P 有稳定模型语义。M 表示了这样的可能状态：明天上街且明天刮风。这与 P 的意图不一致。

下一节介绍的良基语义可以克服稳定模型语义的这些不足。

7. 良基模型语义

良基模型语义（Well Found Model Semantics）是 Van Gelder 等人在 1989 年提出的。目前看来良基模型语义是将理想模型语义或稳定模型语义推广到整个逻辑程序上最为可行的途径。它克服了其它途径的种种不足。同时，Przymusinski 证明良基模型语义等价于四种主要的非单调系统（即 McCarthy 的制限理论，Reiter 的省缺理论，Moore 的自知逻辑和 Reiter 的封闭世界假设）的适当 3-值形式化。

良基模型可以有多种等价的描述。本节采用的是 Przymusinski 提出的迭代最小不动点的定义方法。因为这种方法是最小模型和理想模型的不动点定义的自然扩充，同时与 Fitting 的 Clark 语义扩充有密切联系，再者良基模型语义也可被视为 3-值稳定模型语义。与 Van Gelder 等人提出的原始定义相比，迭代最小不动点定义方法具有良好的构造性。

定义 7.1 （Przymusinski，1989） 设 I，$J\in\mathscr{I}$ 是程序 P 的 3-值解释，A 是一个基原子，$\Theta_J(I)$ 是如下定义的 3-值解释：

（1）$\Theta_J(I)(A)=1$，如果 P 中存在子句 $A\leftarrow L_1,\cdots,L_n$ 使得 $\forall i$，或者 $\hat{J}(L_i)=1$，或者 L_i 是正的，且 $I(L_i)=1$；

（2）$\Theta_J(I)(A)=0$，如果对于 P 中的每一个子句 $A\leftarrow L_1,\cdots,L_n$，存在 L_i（$i\leqslant n$）使得或者 $\hat{J}(L_i)=0$ 或者 L_i 是正的且 $I(L_i)=0$；

（3）$\Theta_J(I)(A)=1/2$，其它。

直观地讲，解释 J 表示当前知道（或为真或为假）的事实，在 $\Theta_J(I)$ 中为真的事实是基于以下假设由 P 一步推出的那些原子，这些假设为在 J 中成立的所有事实和在 I 中为真的所有正事实。在 $\Theta_J(I)$ 中为假的事实是采用封闭世界假设的观点，基于以下假设由 P 一步证明为假的那些原子，这些假设是在 J 中成立的所有事实和在 I 中为真的所有负事实。与第 5 节中扩充的 Van Emden-Kowalski 操作 Ψ_J 不同，Θ_J 以完全对称的方法处理正信息和负信息。

定理 7.1 (Przymusinski, 1989) 对于任一 3-值解释 I，操作 Θ_J 在标准序的意义下是单调的，并且 $\Theta_J^{\uparrow\omega}$ 是 Θ_J 唯一的最小不动点。

我们记 $\Omega_P(J)=\Theta_J^{\uparrow\omega}$。直观地讲，$\Omega_P(J)$ 包含了所有在知道 J 的情况下，采用封闭世界假设的观点，由 P 推导出的所有（为真或为假）的事实。

定理 7.2 (Przymusinski, 1989) Ω_P 存在唯一的 F-最小不动点 M_P，即 $\Omega_P(M_P)=M_P$。进一步，Ω_P 的不动点都是 P 的极小模型。

定义 7.2 (Przymusinski, 1989) 称 Ω_P 的 F-最小不动点 M_P 为程序 P 的良基模型，进而称 M_P 是 P 的良基模型语义。

下面讨论逻辑程序 P 的良基模型的迭代构造。我们定义 P 的解释上的（超穷）序列 $\{I_\alpha\}$：

$I_0=\Omega_P^{\uparrow 0}=\langle\phi;\phi\rangle$；

$I_1=\Omega_P^{\uparrow 1}=\Omega_P(\Omega_P^{\uparrow 0})=\Omega_P(I_0)=\Theta_{I_0}^{\uparrow\omega}$；

$I_{\alpha+1}=\Omega_P^{\uparrow\alpha+1}=\Omega_P(\Omega_P^{\uparrow\alpha})=\Omega_P(I_\alpha)=\Theta_{I_\alpha}^{\uparrow\omega}$；

$I_\delta=\sum_P\{I_\alpha;\alpha<\delta\}$ δ 是极限。

按照前面的分析，$I_{\alpha+1}$ 是在知道 I_α 的情况下由 P 推导出的或为真或为假（在封闭世界假设的意义下）的事实，而这个序列的起点是没有任何信息的状态，即 $\langle\phi;\phi\rangle$，这意味着，$\{I_\alpha\}$ 中的信息均来自程序 P。

可以证明序列 $\{I_\alpha\}$ 是 F-递增的。因为 $\mathscr{H}$ 是可数的，所以 $\{I_\alpha\}$ 中的解释是可数的。因此存在最小的序数 λ 使得 I_λ 是 Ω_P 的不动点，即 $I_{\lambda+1}=\Omega(I_\lambda)=I_\lambda$。这里称 λ 为程序 P 的深度。下面的结论说明了 I_λ 与 P 的良基模型 M_P 之间的关系。

定理 7.4 (Przymusinski, 1989) I_λ 是 Ω_P 的 F-最小不动点，并且与 P 的良基模型 M_P 一致，即：$M_P=I_\lambda=\Omega_P^{\uparrow\lambda}$。

例 7.1 考虑例 5.5 的程序，通过序列 $\{I_\alpha\}$ 构造该程序的良基模型。

$I_0=\langle\phi;\phi\rangle$；

$I_1=\Omega_P(I_0)=\Theta_{I_0}^{\uparrow\omega}$

$\Theta_{I_0}^{\uparrow 0}=\langle\phi;\mathscr{H}\rangle$

$\Theta_{I_0}^{\uparrow 1}=\Theta_{I_0}(\langle\phi;\mathscr{H}\rangle)=\langle\{p(1,2)\};\{p(2,1),p(1,1),p(2,2),q(2)\}\rangle$

$\Theta_{I_0}^{\uparrow 2}=\Theta_{I_0}(\Theta_{I_0}^{\uparrow 1})=\langle\{p(1,2)\};\{p(2,1),p(1,1),p(2,2),q(2)\}\rangle$

可以验证 $\Theta_{I_0}^{\uparrow\omega}=\Theta_{I_0}^{\uparrow 2}$；

$I_2=\Omega_P(I_1)=\Theta_{I_1}^{\uparrow\omega}$

$\Theta_{I_1}^{\uparrow 0}=\langle\phi;\mathscr{H}\rangle$

$\Theta_{I_1}^{\uparrow 1}=\Theta_{I_1}(\langle\phi;\mathscr{H}\rangle)=\langle\{p(1,2),q(1)\};\{p(2,1),p(1,1),p(2,2),q(2)\}\rangle$。

可以验证 $\Theta_{I_1}^{\uparrow\omega}=\Theta_{I_1}^{\uparrow 1}$；

进一步构造可以发现 $I_3=\Omega_P(I_2)=I_2$。所以该程序 P 的良基模型为 I_2，确好是一个 2-值模型。

良基模型与理想模型、稳定模型的关系反映为以下结论。

定理 7.5 (Van Gelder 等，1990) 如果一个程序 P 有 2-值良基模型 M_P，则 P 也有唯一的稳定模型，并且二者是一致的。

推论 7.6 (Van Gelder 等，1990) 局部分层程序的良基模型与其理想模型是一致的。

由于良基模型语义是 3-值意义下的，因此一般情况下它与理想模型语义和稳定模型语义是不同的。例如上节例 6.6 的程序 P 的良基模型为 $M_P=\langle\phi;\phi\rangle$，看上去更符合 P 的直观意图，而 P 的稳定模型为 $\langle\{p,b\};\{a\}\rangle$，二者并不一致。

我们从良基模型与 3-值稳定模型的关系中可以进一步体会良基模型语义的特点。

定理 7.7 一个程序 P 的良基模型 M_P 是 P 的 F-最小的 3-值稳定模型。因此任一逻辑程序至少有一个 3-值稳定模型。进而，良基模型语义与 3-值稳定模型语义是基本一致的。

实际上，P 的良基模型 M_P 可以视为最具怀疑特征的 3-值稳定模型，是关于 P 的可能世界最保守的选择。例如，设程序 $P=\{a\leftarrow\rightarrow b,\ b\leftarrow\rightarrow a\}$，P 有三个 3-值稳定模型：$M_1=\langle\{a\};\{b\}\rangle$，$M_2=\langle\{b\};\{a\}\rangle$，$M_3=\langle\phi;\phi\rangle$，而 P 的良基模型 $M_P=M_3$，显然 M_3 是 P 最保守的 3-值稳定模型。

8. 讨论

逻辑程序语义的研究仍在发展。本文仅涉及了第 2 节 中定义的逻辑程序语义的部

分内容，还没有提及表达形式更丰富的逻辑程序（如带等式的、带或子句的，带否定头子句的逻辑程序）的语义问题。

概括地讲，所谓逻辑程序的（说明性）语义问题就是如何确定或选择一个逻辑程序P指称的模型。一般情况下，P的语义可以表示为$MOD_s(P)$ = {M|M是P的满足语义条件s的模型}。语义条件的不同设定导致P的不同语义规定。这里有两类情况。一是P指称一个模型类，通常$|MOD_s(P)|>1$。例如P的一阶经典语义$MOD_{class}(P)$ = {M|M是P的模型}；P的Clark语义$MOD_{clark}(P)$ = {M|M是comp（P）的模型}；P的Fitting语义$MOD_{Fitting}(P)$ = {M|M是comp（P）的3-值Herbrand模型}；P的Kunen语义$MOD_{Kunen}(P)$ = {M|M是comp（P）的3-值模型}；P的3-值稳定模型语义$MOD_{3\text{-}stable}(P)$ = {M|M是P的3-值Herbrand稳定模型}等等。在这种情况下，我们说P是关于（$MOD_s(P)$所对应的）一组可能世界共性的（不完全）表示。因此，P可以被视为理论。

二是P指称一个特定的模型，即$|MOD_s(P)|=1$，这时$MOD_s(P)$通常用其中仅有的一个模型M_P表示。例如P的最小模型语义M_P^{least}；P的理想模型语义$M_P^{perfect}$；P的稳定模型语义M_P^{stable}；P的良基模型语义$M_P^{well\text{-}found}$等等。在这种情况下，我们说P是（M_P对应的）特定可能世界的（不完全）描述，P仅可被视为模型。

传统上讲，逻辑程序P的说明性语义$MOD_s(P)$是P上一个处理（或推理）过程R的“终审官”。它审查两件事：R的可靠性和完全性。假设P采用R推出的事实A记为$P\vdash_R A$. 说R关于$MOD_s(P)$是可靠的，如果$P\vdash_R A$意味着$MOD_s(P)\models A$。说R关于$MOD_s(P)$是完全的，如果$MOD_s(P)\models A$意味着$P\vdash_R A$。由于前面各节介绍的各种试图反映常识背景的模型语义的提出，完全性的原则首先受到挑战。一方面，从理论上讲，对于某种模型语义$MOD_s(P)$的规定，可能根本不存在完全的R。例如Fitting语义就是如此。另一方面，从实践上看，考虑到能行可计算问题，要求R是完全的可能在实践上也行不通。实际上几乎所有的Prolog系统均放弃了完全性的原则。

由于$MOD_s(P)$通常是$MOD_{class}(P)$的真子集，关于$MOD_s(P)$的可靠性原则已经包容了许多非逻辑可靠的推理，例如几种常见的非单调推理。无论从理论上还是从实践上讲，推理关于$MOD_s(P)$的可靠性原则仍被普遍遵循。但是随着人们对归纳、类比学习等认识进程的研究，可靠性原则也受到了挑战。例如，假设P_I为：

natural-number (0)；

natural-number (s (0))；

natural-number (s (s (0)))。

P_I采用最小模型语义$M_{P_I}^{least}$，作用于P上的R具有归纳和演绎的能力：R可以从P_I中归纳出规则：

natural-number (s (x)) ←natural-number (x)

由此演绎出natural-number (s (s (s (0))))，即

$P_I\vdash_R$ natural-number (s (s (s (0))))。

显然R不是关于$M_{P_I}^{least}$可靠的。因为

$M_{P_I}^{least}\not\vdash$ natural-number (s (s (s (0))))。

我们认为知识表示的逻辑途径给AI带来的束缚在一定意义上源于“可靠性”和“完全性”两条传统原则。逻辑的基本目标是研究可靠完备的推理，可靠性和完全性自然是必须坚持的原则。但知识表示的基本目的是表达问题域中的可能世界，并在此基础上进行问题求解，“可靠性”和“完全性”原则常常变得不那么中肯了。我们更主张逻辑程序P仅代表一个可能世界，P的功能语义取决于在P所表达的可能世界上所能完成的问题求解。也就是说，P的说明性语义不再作为P上的处理过程的“终审官”，而仅仅是实现P的功能语义（使用P所进行的问题求解操作的全体）的基本依据或素材。好比当前人们关于现实世界的可能模型仅仅是新的认识

计算机科学 1994Vol. 21 No. 2

代数规范描述及其语义的研究*)

宋 群 聂承启

(江西师范大学计算机科学系 南昌 330027)

摘 要 In this paper, the theories and the concepts of algebraic specification and its semanteme for specification abstract data type are presented. The semantic basis and semantic meaning of algebraic specification are discussed. In the last part of this paper, some examples of modeling abstract data type defined accurately are given.

关键词 Algebraic Specification, Semantic Function, Algebraic Class Abstract Data type

用代数规范描述来描述抽象数据类型的基本思想是用它的标记和特征性质来说明抽象数据类型，它们的性质可用多类逻辑形式来表示，通常为受限的一价逻辑，例如等式逻辑。但也可以选用其它形式，象高阶等式逻辑等。抽象数据类型被刻画为计算结构的同构类。因此，从数学上来说，代数规范描述的理论就是可达多类代数的同构类的理论。

抽象数据类型规范的方法研究主要有二种：一是从给定的数据类型D着手，去找一个适合D的代数规范描述。二是从一个代数规范SP开始(比如对一些给定的非正式的软件或硬件问题)，试图去获得一个更详细的规范描述SP'来描述一个抽象数据类型。本文中，我们主要兴趣在后一种方法的研究。

1 公式与理论

多类代数的性质是通过多类一阶等式逻辑来表示的。

设$\Sigma=(S, F)$是一个标记，一个Σ-等式具有形式$t=_s t'$，其中$t, t' \in T(\Sigma, X)$是类$s\in S$中的项。(一阶)Σ-公式是Σ-等式由连接词$\neg$，$\wedge$和量词$\forall$，$\exists$构造而成。Σ-公式集WFF(Σ)是满足以下性质的最小集合：

(Ⅰ) 每个Σ-等式在WFF(Σ)中；

(Ⅱ) 若G，H$\in$WFF(Σ)，则$\neg$G，(G$\wedge$H)$\in$WFF(Σ)；

(Ⅲ) 若$x\in Xs$且$G\in$WFF(Σ)，那么$(\forall x_s . G)\in$WFF(Σ)；

*) 江西省自然科学基金资助课题。聂承启 副教授，从事软件工程、知识工程、多处理系统和计算机应用基础的研究。92年在美国Oakland大学研究多处理系统。宋群 讲师，从事软件形式化开发方法的研究，现在德国攻读博士学位。

过程的基点，认识起点不应限制由此展开的认识过程。逻辑程序语义可以作为它所表达的认识进程$\{P_i\}$实际指称的轨迹——$\{M^*_{P_i}\}$。从这个角度上看，我们更有理由接受那种"可想而不可及"的模型语义定义：这类语义模型(如稳定模型语义、良基模型语义)尽管可能无法有穷(或能行)生成，但确是良定的。(全文完)

参 考 文 献

[1] H. Przymusinska & T. Przymusinski, Semantic Issues in Deductive Databases and Logic Programs. Formal Techniques in Artificial Intelligence, R. B. Banerji (editor), 1990, pp. 321-367.

[2] J. W. Lloyd, Foundations of Logic Programming, Springer Verlag, New York, N. Y., first edition, 1984.