第9卷第4期
1998年4月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 9, No. 4
Apr. 1998

处理知识库中不一致性的超决定逻辑研究*

王清毅 陈恩红 刘贵全 蔡庆生

（中国科技大学计算机系 合肥 230027）

摘要 本文首先分析了现有的能够处理知识库中不一致性的逻辑的不足，然后定义了超决定结构、模型和语义的概念，具体描述了超决定语义的计算过程，从而给出了一种能够处理知识库中不一致性的逻辑，并对该逻辑进行了一些讨论.

关键词 知识库，不一致性，超决定逻辑.

中图法分类号 TP18

我们知道，如果在一个由许多专家提供知识的知识系统中基于经典逻辑进行推理，则由于知识源（专家）之间可能存在着不一致性（Inconsistency）和经典逻辑的平凡性（Triviality），就会推出任意结论. 例如，在一个应用于天气预报的知识系统中，由于其推理机制的平凡性，就会推出任意结论，这显然不是我们所需要的. 在人的常识推理中，这种知识之间的不一致性随处可见，推理是在知识不一致或不完全的背景下进行的. 一个人可能会对某个人或事物存在矛盾的看法，但这并不影响其进行合理的推理和行为. 在计算机应用中，我们不希望所建造的计算机智能系统由于其知识库中存在着局部的不一致就会导致整个系统变得毫无用处（可推出任意结论）. 因此，知识库中的不一致问题是基于知识的推理所面临的一个基本问题. 在人工智能研究中，这类问题也称为超协调问题.[1,2]

1 问题的提出

例1：以下是一个某大学招聘教师所需条件的知识库：

engaged(X)←has_papers(X,Y)∧has_doctorate(X)∧has_research_projects(X,national_level)∧Y≥5

engaged(X)←has_papers(X,Y)∧has_doctorate(X)∧has_research_projects(X,provincial_level)∧Y≥5

engaged(X)←has_papers(X,Y)∧¬has_doctorate(X)∧has_research_projects(X,national_level)∧Y≥10

has_papers(liming,10)←

has_research_projects(liming,national_level)←

知识库中的知识是一致的. 但其中的事实并未说明 liming 是否拥有博士学位. 经典逻辑中的推理过程如下：如果 liming 拥有博士学位，则他满足第1个子句的体，可以被招聘；如果 liming 不拥有博士学位，那么他满足第3个子句的体，他也可以被招聘. 即不论李明是否拥有博士学位，都可以推出 engaged(liming)，这里实际上隐含运用了经典逻辑中的排中律. 但在 Gelfond 和 Lifschitz[3]及 Kowalski 和 Sadri[4]所提出的逻辑框架中都无法推出 engaged(liming)，在 Blair 的标记逻辑框架中也不能推出 engaged(liming):t.

例2：下面是一个表示一些省地理位置关系的知识库：

south_neibouring(hebei,shandong)←　　south_neibouring(shandong,anhui)←

south_neibouring(anhui,zhejiang)←　　south_neibouring(zhejiang,fujian)←

south_neibouring(fujian,shandong)←　　on_the_north_of(X,Y)←south_neighbouring(X,Y)

on_the_north_of(X,Y)←on_the_north_of(X,Z)∧on_the_north_of(Z,Y)−¬on_the_north_of(X,X)←

neighbouring(neimeng,hebei)←　　neighbouring(jiangsu,shandong)←

neighbouring(jiangxi,anhui)←　　neighbouring(jiangxi,zhejiang)←

---

* 本文研究得到国家自然科学基金和高等学校博士点基金资助. 作者王清毅，1962年生，博士生，主要研究领域为机器学习，知识发现. 陈恩红，1968年生，博士，讲师，主要研究领域为机器学习，知识发现，遗传算法. 刘贵全，1970年生，博士生，主要研究领域为人工智能，机器学习，知识发现. 蔡庆生，1938年生，教授，博士导师，主要研究领域为人工智能，机器学习，离散数学，知识发现.

本文通讯联系人：王清毅，合肥 230027，中国科技大学计算机系

本文1996-12-04收到原稿，1997-05-13收到修改稿

显然,这个知识库是不一致的.因为从前5条事实和2条规则可以推出相互矛盾的结果.以这个知识库作为背景知识,Gelfond 和 Lifschitz 所提出的逻辑就不能进行正确的推理.例如,neighbouring(neimeng,fujian)就会被推出,这显然是错误的;而按照 Kowalski 和 Sadri 所作的否定知识优先于肯定知识的假设,就会推出河北不南邻山东,山东不南邻安徽,安徽不南邻浙江,浙江不南邻福建,福建不南邻山东.而事实上,除福建南邻山东这一条事实错误外,其他4条事实都是正确的.

从上面两个例子可以看到:在许多情形下,目前的能够处理知识库中不一致性的逻辑在一致的知识库和不一致的知识库中都不能推出正确的结果.这是因为在这些逻辑中都假排中律无效、不能分情况推理(Reasoning by Cases)以及不能阻止不一致性(矛盾)的传播.[5]具体说来,这些逻辑仍有3个问题需要解决:(1) 满足排中律的推理.即如果 A 蕴含 B,$\neg A$ 也蕴含 B,则可以在不确切地知道 A 和 $\neg A$ 哪一个为真的情形下推出 B;(2) 分情况推理.即如果已知 B 或 C 为真,且已知 B 蕴含 D,C 蕴含 D,则可以在不知道 B 和 C 哪一个为真的情形下推出 D;(3) 阻止不一致性(矛盾)的传播.即某些局部的不一致不应该使得整个知识库变得无法使用.

本文从满足这3个特性出发,定义了超决定(Over-determined)结构、模型和语义的概念,具体描述了超决定语义的计算过程,从而给出了一种能够处理知识库中不一致性的超决定逻辑.

2 超决定逻辑的定义

设 ξ 是一个由无限个变元,有限个常量、函数和谓词构成的逻辑语言.

定义1. 如果 $L_0,L_1,\ldots,L_n$ 是文字(正文字或负文字),那么称 $L_0\leftarrow L_1\wedge\ldots\wedge L_n$ 为一般 Horn 子句.通常子句中的变元被认为在子句前被全称限定.

定义2. 一个一般 Horn 程序(简称 GHP)是一个一般 Horn 子句(可以为无限个)的集合.

我们约定所讨论的知识库由一般 Horn 程序构成,GHP 可以表示不一致信息,举例如下.如果 P 是一个一致的 GHP,我们可以向 P 中加入以下两个子句而构成 Q:$q\leftarrow$,$\neg q\leftarrow$.其中 q 是不在 P 中出现的命题符号.显然,Q 是不一致的.尽管 Q 的不一致性仅反映在这两个新加入的子句上,但在现有的经典逻辑系统面前,Q 也是没有意义的.而本文给出的逻辑,旨在使得这种局部的不一致性不影响 Q 的其余部分.换句话说,基于不一致知识库 Q 仍能进行正确地推理.

我们知道,一个语言 ξ 的 Herbrand 结构 I 是 ξ 的 Herbrand 基的一个子集,即 $I\subseteq B_\xi$,其中 B_ξ 是一个 ξ 的 Herbrand 基.记 $I^*=I\cup\{\neg a|a\notin I\}$,即 I^* 由向 I 中加入所有不在 I 中的原子的否定所构成,这可以视为在否定没有被明确指定时的一般意义下的 I.由此,可以说一个 Herbrand 结构 I 是语言 ξ 的基文字集合的子集且使得对所有基原子 $a\in B_\xi$,I 仅含有 $\{a,\neg a\}$ 中的一个.如果定义 I 至少含有 $\{a,\neg a\}$ 中的一个,我们就得到了一个超决定结构 I.

假设 I 是一个超决定结构,即 I 是一个基文字的集合使得对每一个 $a\in B_\xi$,$\{a,\neg a\}$ 中至少有一个 $\in I$.我们仍然使用经典逻辑意义下的否定$\neg$、合取$\wedge$、析取$\vee$和蕴涵$\leftarrow$.我们定义超决定逻辑的逻辑后承$\models_{od}$如下(od 代表 Over-determined的缩写):

$$I\models_{od}L_1\vee\ldots\vee L_n,\quad \text{当且仅当 } L_1\in I,\ldots,\text{或 } L_n\in I.$$

定义超决定逻辑中蕴涵式 $F\leftarrow G$ 成立的条件为 $I\models_{od}F\leftarrow G$,当且仅当 $I\models_{od}F$ 或 $I\not\models_{od}G$.假定 P 是一个不一致的理论,称 I 是一个 P 的超决定模型,当且仅当对每个公式 $F\in P$,$I\models_{od}F$.P 的超决定语义是所有基析取式(由正文字和负文字组成)d 的集合,记为 Γ,使得对 P 的每一个超决定模型 I,$I\models_{od}d$,从而 $P\models_{od}d$.超决定一词的含义是指在知识不一致的背景下仍然可以进行正确的推理.

例3:考虑一个 GHP P:$b\leftarrow a$,$\neg b\leftarrow a$.

按照经典逻辑,$\neg a$ 是 GHP 的推论.但从超决定逻辑的角度来看,这个 GHP 有3个最小化超决定模型,它们是:$I_1=\{a,b,\neg b\}$,$I_2=\{b,\neg a\}$,$I_3=\{\neg b,\neg a\}$.可见,$P\not\models_{od}\neg a$.

例4:让我们再看一看在超决定逻辑中例1的结果.

设 I 是例1中知识库的任一个超决定模型,则 I 必含有 has_doctorate(liming)或$\neg$has_doctorate(liming).若 has_doctorate(liming)$\in I$,则 I 满足第1个子句的体.但 I 是超决定模型,所以,必有 engaged(liming)$\in I$;若$\neg$has_doctorate(liming)$\in I$,则 I 满足第3个子句的体,同样有 engaged(liming)$\in I$.因此,engaged(liming)在 P 的任一模型中都为真,所以必有 $P\models_{od}$engaged(liming).

定理1. 设 P 是一个 GHP,A,B,C 是关于基原子的元变量,则有以下公理:

(1) 排中律.$A\vee\neg A$

(2) $(A \vee B) \leftarrow (A \leftarrow \neg B)$

(3) 分情况推理. $A \leftarrow ((A \leftarrow B) \wedge (A \leftarrow C) \wedge (B \vee C))$

(4) 对不同于 A 的 B, $(A \wedge \neg A) \not\models_{od} B$. 当然, $(A \wedge \neg A) \models_{od} A$ 和 $(A \wedge \neg A) \models_{od} \neg A$. 这表明在超决定逻辑中,矛盾被局部化在自身,从而排除了推理的随意性.

证明:设 I 是一个超决定结构,

(1) 对任一原子 A, $A \in I$ 或 $\neg A \in I$, 因此 $I \models_{od} A \vee \neg A$.

(2) 假定 $I \models_{od} (A \leftarrow \neg B)$ 和 $I \not\models_{od} A$, 这样有 $I \models_{od} B$. 因为 $I \not\models_{od} A$, $I \models_{od} (A \leftarrow \neg B)$, 必有 $I \not\models_{od} \neg B$. 因此, $\neg B \notin I$. 所以 $B \in I$, 即有 $I \models_{od} A \vee B$.

(3) 假定 $I \models_{od} (A \leftarrow B) \wedge (A \leftarrow C) \wedge (B \vee C)$, 则有 $I \models_{od} B$ 或 $I \models_{od} C$. 若 $I \models_{od} B$, 因为 $I \models_{od} (A \leftarrow B)$, 所以 $I \models_{od} A$; 若 $I \models_{od} C$, 因为 $I \models_{od} (A \leftarrow C)$, 所以 $I \models_{od} A$.

(4) 取 $I = \{a, \neg a, \neg b\}$, 显然 I 是子句 $a \leftarrow$ 和 $\neg a \leftarrow$ 的超决定模型, 但 $I \not\models_{od} b$. □

值得注意的是,在定理的第(2)部分,在超决定逻辑中, $A \vee B$ 并不等价于 $A \leftarrow \neg B$. 即我们虽有 $(A \vee B) \leftarrow (A \leftarrow \neg B)$, 但却没有 $(A \leftarrow \neg B) \leftarrow (A \vee B)$. 例如, $I = \{\neg A, B, \neg B\}$ 满足 $A \vee B$, 但却不满足 $A \leftarrow \neg B$. 这是因为我们在超决定逻辑结构中重新定义了蕴涵式成立的条件.

3 超决定语义的计算

设 S 是一个基子句集. 超决定语义的计算采用了3种类型的变换规则,即

(1) 展开规则. S 的展开为: $unfold(S) = \{L \leftarrow B_1 \wedge D \wedge B_2 \mid L \leftarrow B_1 \wedge L' \wedge B_2 \in S, L' \leftarrow D \in S\}$

(2) 配对规则. S 的配对为: $pair(S) = \{L \leftarrow Q_1 \wedge Q_2 \mid L \leftarrow R_1$ 和 $L \leftarrow R_2 \in S$, 且 $A \in R_1$, $\neg A \in R_2$, $Q_1 = R_1 - \{A\}$, $Q_2 = R_2 - \{\neg A\}\}$.

(3) 否定消除规则. S 的否定消除为: $ne(S) = \{L \leftarrow Q \mid L \leftarrow R \in S, R = Q \cup \{\neg L\}\}$.

前已述及,我们讨论的子句允许否定形式在子句的头和体中出现. 因此就有可能面临GHP中出现不一致,而配对规则和否定消除规则正是为了处理子句中出现的否定而引入的.

定义 3. 给定一个GHP P, 定义 P 的变换 $\Gamma(P)$ 为

$$\Gamma(P) = P \cup unfold(P) \cup pair(P) \cup ne(P).$$

P 的迭代变换 $\Gamma_\infty(P)$ 为

$$\Gamma_0(P) = grd(P) \qquad (grd(P)\text{表示 } P \text{ 中子句的所有基例所构成的集合})$$

$$\Gamma_{i+1} = \Gamma(\Gamma_i(P))$$

$$\Gamma_\infty(P) = \bigcup_{i=0}^{\infty} \Gamma_i(P).$$

命题 1. 假定基子句 $C_1: L \leftarrow B_1 \wedge L' \wedge B_2$, $C_2: L' \leftarrow D$, 则基子句 $C: L \leftarrow B_1 \wedge D \wedge B_2$ 是 C_1 和 C_2 的超决定语义推论(证明略).

命题 2. 假定基子句 $C_1: L \leftarrow R_1$, $C_2: L \leftarrow R_2$, 且存在一个原子 A, $A \in R_1$ 且 $\neg A \in R_2$. 设 $Q_1 = R_1 - \{A\}$, $Q_2 = R_2 - \{\neg A\}$, 则基子句 $C: L \leftarrow Q_1 \wedge Q_2$ 是 C_1 和 C_2 的超决定语义推论(证明略).

命题 3. 假定 C 是基子句: $L \leftarrow \neg L \wedge Body$, 则基子句 $C': L \leftarrow Body$ 是 C 的超决定语义推论(证明略).

命题 4. 假定 $P \models_{od} L$, 则 $P \vdash L$. 进一步设 P' 是 P 的一致子集, 则 $P \models_{od} L$, 当且仅当 $P \models L$.

证明:这是因为每一个二值逻辑中的解释 I 可以视为一个超决定结构 I.

命题4表明超决定语义推论也是经典逻辑的语义推论;反之,经典逻辑的语义推论未必就是超决定语义推论. 这在例2中已经说明了. 超决定逻辑对于经典逻辑来说是合理的,但却不是完备的. 我们说经典逻辑在不一致的知识库中的推理具有随意性. 而超决定逻辑却可以很好地解决这一问题. 在知识库是一致的或知识库是不一致的,但却不直接影响的情况下,超决定逻辑等价于经典逻辑. 因此,从这个角度,我们可以说超决定逻辑是经典逻辑的又一扩展. 由命题1、命题2和命题3,通过归纳法不难证明以下定理成立.

定理 2. $\Gamma_\infty(P)$ 是 P 的超决定语义推论, 即 $P \models_{od} \Gamma_\infty(P)$.

命题 5. 如果 $P \models_{od} L$, 那么在 P 中一定存在一个以 L 作为头的子句(证明略).

下面我们给出超决定语义的计算过程. 为此先给出相关的定义和引理.

定义 4. 给定两个具有同样的头的基子句 C 和 C', 称 C 将 C' 归类, 当且仅当 C 的体中的文字集合是 C' 的体中的

文字集合的子集.

定义 5. 假定 P 和 P' 是 GHP,称 P 归类 P',如果对 P' 中的每一个子句 C',P 中都有一个子句 C 将 C' 归类.

引理 1. 如果 P 将 P' 归类,那么 $\Gamma_\infty(P)$ 将 $\Gamma_\infty(P')$ 归类(证明略).

定理 3. 假定 P 是一个没有函数的逻辑程序,则 $P\models_{od}L$,当且仅当 $L\leftarrow\in\Gamma_\infty(P)$.

证明:充分性:如果 $L\leftarrow\in\Gamma_\infty(P)$,那么由定理 2 得到 $P\models_{od}L$.

必要性:假定 $P\models_{od}L$.设 $P'=\Gamma_\infty(P)$.

首先,从 P' 中消去两种类型的子句.第 1 种,子句 C 的头也出现在子句 C 的体中,这时子句 C 实际上是一个永真式;第 2 种,如果 P' 中的一个子句 C 被 P' 中的另一个子句 C' 归类,则从 P' 中删去子句 C.特别要注意的是,当 L 是子句头,$\neg L$ 出现在子句体中时,由否定消除规则可以从这样的子句体中消去 $\neg L$ 而形成一个归类原子句的子句,结果是将原子句删除.其次,再从 P' 中除去冗余的文字,即如果 K 出现在一个子句的体中,且 $K\leftarrow\in\Gamma_\infty(P)$,则由本定理的充分性证明部分可以推知 K 是冗余的,应从子句中删去 K.

显然,在删除了冗余的子句和文字后的 P',在超决定语义中仍然和 $\Gamma_\infty(P)$ 是逻辑等价的.以 $P'(L)$ 表示 P' 中以 L 作为头的子句的集合.显然 $P'(L)$ 不为空集(命题 5)且 $P'(L)$ 中所有子句的体中都不含有 L,$\neg L$ 及文字 $K(K\leftarrow\in\Gamma_\infty(P))$.设基原子 A 或 $\neg A$ 出现在 $P'(L)$ 中一个子句的体中,记 X 为所有这样的基原子 A 的集合,设 $X=\{A_1,\ldots,A_m\}$.设 $X'=X\cup\{\neg A\mid A\in X\}$,称 X' 的一个子集 S 为一个 X-候选.如果对每一个 $A\in X$,$\{A,\neg A\}$ 中仅有一个 $\in S$,可记 $S=\{A_1^*,\ldots,A_m^*\}$,其中 A_i^* 为 A_i 或 $\neg A_i$.称一个 X-候选为超决定可扩张的,如果可以向该 X-候选中添加不在 X' 中的文字而使该 X-候选扩张成一个 P' 的超决定模型.设 n 是不能被超决定扩张的 X-候选的个数,考虑 X 保持不变,我们通过对 n 进行归纳来证明.

基本情形 $n=0$:设 $B_1,\ldots,B_r$ 是 $P'(L)$ 中所有子句的体,称 B_i 覆盖一个 X-候选 S,如果 B_i 中文字的集合是 S 的一个子集.如果 B_i 覆盖 S,则 S 的每一个超决定扩张模型必含有 L.这是由 B_i 和超决定模型本身的定义所决定的.这时又有两种情形:

情形 1.所有 X-候选都被一些 B_i 所覆盖($1\leqslant i\leqslant n$).将子句 $L\leftarrow X_j$ 添加到 $P'(L)$ 中而将 $P'(L)$ 扩张成 $P^*(L)$,其中 X_j 是一个 X 候选.注意到 $P'(L)$ 将 $P^*(L)$ 归类,所以由引理 1,$\Gamma_\infty(P'(L))$ 将 $\Gamma_\infty(P^*(L))$ 归类.$\Gamma_\infty(P^*(L))$ 含有所有 $L\leftarrow C_j$ 形式的子句,这里 C_j 是一个 $(X-A_m)$-候选.为证明这一点,假定 $L\leftarrow A_1^*\&\ldots\&A_{m-1}^*$ 是这样的一个子句,其中 A_i^* 为 A_i 或 $\neg A_i$.下列两个子句必在 $P^*(L)$ 中:(1) $L\leftarrow A_1^*\wedge\ldots\wedge A_{m-1}^*\wedge A_m$,(2) $L\leftarrow A_1^*\wedge\ldots\wedge A_{m-1}^*\wedge\neg A_m$.由此,根据配对规则,$\Gamma_\infty(P^*(L))$ 中必含有子句 $L\leftarrow A_1^*\wedge\ldots\wedge A_{m-1}^*$.可见对每一个 X-候选,可以从 X 候选中消去 A_m 而得到 X_{m-1}-候选.我们可以重复上述过程,使用类似子句(1)(2)的子句以消去 A_{m-1}.依此类推,重复这一变换过程 m 次,我们可推出单元子句 $L\leftarrow$ 必在 $\Gamma_\infty(P^*(L))$ 中.再由引理 1,且因为只有 $L\leftarrow$ 将其本身归类,所以 $L\leftarrow$ 也必在 $\Gamma_\infty(P^*(L))$ 中.又由于 P' 和 $\Gamma_\infty(P)$ 在超决定语义下是逻辑等价的,即它们有着相同的超决定模型且 $L\leftarrow\in(P'(L)$ (定理 2),因此必有 $L\leftarrow\in P'$,$L\leftarrow\in\Gamma_\infty(P)$.

情形 2.存在一个 X-候选不被任何 B_i 所覆盖($1\leqslant i\leqslant n$).可以证明在这种情形下,存在一个不含有 L 的 P' 的超决定模型.但这又和假设 $\Gamma_\infty(P)=_{od}L$ 相矛盾.具体证明过程可以通过对 w(w 为不被任何 B_i 所覆盖的 X-候选的个数)进行归纳而完成.限于篇幅,不再详述.

归纳情形 $n=k+1$:假设 Q 为一个理论,X 仍然和 $n=0$ 时相同.设有 $n=k$ 个 X-候选不能被超决定扩张,如果 $Q\models_{od}L$,那么 $L\leftarrow\Gamma_\infty(P)$.设 $S=\{L_1,\ldots,L_m\}$ 是 P' 中一个不能被超决定扩张的 X-候选,从 P' 中删去所有形式为 $\neg L_i\leftarrow Body$ 的子句,其中 $Body$ 不含有任何 $\neg L_j(L_j\in S)$,这样 $Body$ 一定是 S 的一个子集.P' 中至少存在一个这种形式的子句,否则,我们能通过向 S 中添加所有不在 X' 中的文字而将 S 扩张成一个超决定模型.这一点可由 S 和 S 的扩张超决定模型的定义共同推出.向 Q 中加入新子句 $L\leftarrow L_1\wedge\ldots\wedge L_m$.对 Q 来说,S 现在是可以超决定扩张的.例如,设 $M=S\cup\{K\mid K$ 是一个文字,$K\notin X\}$,则 M 就是 S 的一个超决定模型.这是因为所有可能使得 M 不能成为一个超决定模型的形式为子句 $\neg L_i\leftarrow Body$ 的子句都已从 P' 中删除.子句 $L\leftarrow L_1\wedge\ldots\wedge L_m$ 的加入并没有影响超决定可扩张性,这是因为每一个 P' 中的可扩张的 X-候选在 Q 中仍然是超决定可扩张的.因此,Q 仍然满足归纳假设,所以 $L\leftarrow\Gamma_\infty(Q)$.将 $L\leftarrow$ 归入 $\Gamma_\infty(Q)$,这时有两种情形:第 1,不需要以上新子句 $L\leftarrow L_1\wedge\ldots\wedge L_m$,则 $L\leftarrow\in\Gamma_\infty(P)$;第 2,需要该新子句,则所有形式为 $L\leftarrow\neg L_1,\ldots,L\leftarrow\neg L_m$ 一定是在 Q 中或可由 Q 中导出,因此,这些子句也一定是在 P' 中.通过展开规则的运用,可以推知子句 $L\leftarrow Body$ 一定是在 P' 中或可被 P' 中一个形式为 $L\leftarrow Body'$ 的子句归类.由 X' 的定义知,$Body'$ 一定是 X' 的一个子集,但 $Body'$ 又是 $Body$ 的子集,所以 $Body'$ 不含有 $\neg L_j(L_j\in S)$,$Body'$ 也一定是 S 的一个子集.既然 $Body$ 和 $Body'$ 都是 S 的子集,则通过 P' 中的子句 $L\leftarrow\neg L_i$ 和 $L\leftarrow Body$ 或 $L\leftarrow Body'$,再由配对规则,可

以得到 $L\leftarrow P'$，所以 $L\leftarrow\in\Gamma_\infty(P)$. □

以上证明过程实际上也给出了超决定语义的计算过程，即通过反复运用展开、配对和否定消除 3 个规则从 P 构造出 $\Gamma_\infty(P)$，直到没有新的子句出现. 这时知识库中的单元子句的集合就是可从 P 中推出的基文字的集合. 换句话说，当需要从知识库中推出一个文字 L 时，就是反复运用这 3 条规则，直到 $L\leftarrow$ 出现.

4 讨 论

知识不一致背景下的推理是基于知识推理的一个重要研究课题. 一致性知识常常是可望而不可及的. Gelfond 和 Lifschitz 已经提出逻辑程序设计应该容许两种否定出现在程序中，即经典的否定和作为失败的否定. 然而一旦不一致性出现在 Gelfond Lifschitz 语义中，这种不一致性的传播和相应的公式变得立即可以推出. 这样往往导致整个知识库中的错误知识的增加，难以进行正确的推理. Kowalski 和 Sadri 所提出的逻辑程序设计的语义虽然允许知识库中包含上述两种否定，但当知识库中存在不一致性的时候，该语义总是选择否定性知识，摒弃肯定性知识. 而在许多情形下，这种处理方法也是不很恰当的. 这是因为这些逻辑都使得排中律无效，而我们知道使排中律无效来处理知识的不一致性没有实质意义[2]，相反却使得一些本来能在经典逻辑中推出的结果在这些逻辑中却不能推出.

从技术途径上看，超决定逻辑没有象其他能处理知识不一致性的超协调逻辑和容错逻辑那样，从破坏经典逻辑中的传递律、合取律和等价律出发来构造，而是在扩展的 Herbrand 结构范围内定义了超决定逻辑后承关系和蕴涵式成立的条件. 但超决定逻辑也是通过限制矛盾（不一致性）来维护知识状态的相容性（一致性），这是和超协调逻辑和容错逻辑的共同之处. 如果考虑将超决定逻辑后承 $\models_{od}$ 和经典逻辑后承 $\models$ 相对应，将 $P\models_{od} B\leftarrow A$ 和经典逻辑中的命题 $\{\neg p\}\models q\leftarrow p$ 和 $\{q\}\models q\leftarrow p$ 相对应，则我们不难将经典逻辑中的一些结果移植到超决定逻辑中来. 定理 1 的(1)和(3)表明，当需要使排中律有效和能分情况进行推理时，超决定逻辑是更为适合的，(4)表明超决定逻辑也是含矛盾但不平凡的理论. 本文定义了超决定结构、模型和语义的概念，具体描述了超决定语义的计算过程，给出了一种处理知识库中不一致性的新的逻辑和方法. 超决定语义计算方法的效率和具体实现的数据结构则是一个有待研究的课题.

参考文献

1 林作铨，李未. 超协调逻辑(I)——传统超协调逻辑研究. 计算机科学，1994，21(5)：1～4
(Lin Zuo-quan, Li Wei. Over-consistent logic (I)——research on conventional over-consistent logic. Computer Science, 1994, 21(5): 1～4)

2 林作铨，李未. 超协调逻辑(II)——新超协调逻辑研究. 计算机科学，1994，21(6)：1～40
(Lin Zuo-quan, Li Wei. Over-consistent logic (II)——research on current over-consistent logic. Computer Science, 1994, 21(6): 1～4

3 Gelfond M, Lifschitz V. Logic programs with classical negation. In: Proceedings of the 7th International Conference on Logic Programming. Cambridge, MA: MIT Press, 1990. 579～597

4 Kowalski R, Sadri F. Logic programs with exceptions. In: Proceedings of the 7th International Conference on Logic Programming. Cambridge, MA: MIT Press, 1990. 598～613

5 Grant J, Subrahmanian V S. Reasoning in inconsistent knowledge bases. IEEE Transactions on Knowledge and Data Engineering, 1995, 7(1): 177～189

Research on Over-determined Logic for Handling the Inconsistency in Knowledge Bases

WANG Qing-yi　CHEN En-hong　LIU Gui-quan　CAI Qing-sheng

(*Department of Computer Science　University of Science and Technology of China　Hefei　230027*)

Abstract　In this paper, the authors discuss some drawbacks of several kinds of existing logic to handle inconsistency in knowledge bases at first, and then give the definition of over-determined structure, over-determined model and over-determined semantics. They also describe in detail the computation procedure for over-determined-semantics. As a result, a kind of logic for handling inconsistency is presented. Finally, a discussion on the over-determined logic is concluded with.

Key words　Knowledge bases, inconsistency, over-determined logic.