评价目标前提的逻辑框架

夏兰亭[1] 韩道军[1] 边 芮[2] 李 磊[1]
(中山大学软件研究所 广州 510275)[1] (广东商学院公共管理学院 广州 510320)[2]

摘 要 多前提决策问题是决策领域中重要的研究内容,其决策质量可反映决策理论研究的成果。在多前提决策中,目标会有多种实现方式,每种实现方式都有相应的实现难度。在多前提决策研究中,定义了目标的条件集,提出了将条件集分解到最简单形式的算法,并证明了算法的输出为目标的极小最简不完备条件集。然后,提出了条件集可信度的概念及其计算方式,利用可信度评估目标的所有极小最简不完备条件集的实现难度,并以此确定实现该目标的最优条件集。最后,给出了评价目标前提的逻辑框架O的形式定义,证明了框架O的可计算性和推理能力,并用逻辑程序Prolog实现了框架O的原型,通过对框架O的原理性实验和具体的工程应用验证了框架O的有效性。

关键词 多前提,最优条件集,可信度,实现难度,评价
中图法分类号 TP301 **文献标识码** A

Logical Framework for Evaluation of Multiple-premise

XIA Lan-ting[1] HAN Dao-jun[1] BIAN Rui[2] LI Lei[1]
(Software Research Institute, SunYat-sen University, Guangzhou 510275, China)[1]
(School of Public Management, Guangdong University of Business Studies, Guangzhou 510320, China)[2]

Abstract The multi-premise problem is an important research field of decision making and its optimal decision reflects the state of the art research of decision theory. Many methods can achieve it in multi-premise decision making, and each method has its own difficulties. This paper first introduced a new concept-premise set, proposed an algorithm to decompose premise set into the simplest form, and proved that the algorithm outputs minimal, non-complete premise sets for any given goal. Next the credibility of each premise was defined for measuring the difficulty of premise set, and the optimal premise set was recommended for the goal. Then this paper gave a formal description of the logical framework O for evaluation of multiple-premise, and proved its computability and reasoning faculty. Finally, a prototype of framework O was implemented in Prolog and its soundness was proved via experiments and application in software engineering.

Keywords Multi-premise, Optimal premise set, Credibility, Difficulty of premise set, Evaluation

1 引言

多前提决策问题[1-3]是决策领域中的重要研究内容,其决策质量体现了决策理论研究的成果,许多优秀的专家系统、咨询系统都会涉及对多前提决策问题的研究。在多前提决策问题的研究中,实现目标一般会有多种方式,每种实现方式都有若干个前提条件。如果某个实现方式的所有前提条件都被满足(完备的),那么可用该实现方式实现目标,否则当前目标不能用该方式实现。在多前提决策中,选择最优的实现方式是其研究的核心问题。寻找最优解的基本方法主要有基于专家系统的决策方法和基于概率的决策方法。

E. A. Feigenbaum等人最早提出了领域专家的概念,并模拟专家求解问题的思维过程,实现了化学领域专家系统DENDRAL,然后经过C. Mellon, A. Newell和H. Simon等人的努力,使专家系统的研究与应用走向成熟[4]。领域专家系统的基本特征[3-5]是:系统拥有一个解决问题的知识库。给定一个待解决的目标和该目标所有实现方式的知识。如果某个实现方式的所有前提条件都可由知识库推出,那么知识库就可用该方式实现目标。如果目标可由多种方式实现,那么把实现该目标所涉及的相关知识进行加权评估,以选择最优的实现方式。在现实中,我们不一定对目标拥有完备的知识。当知识不完备时,领域专家系统的知识库未必能推出某个实现方式的所有前提条件。如果没有一个实现方式可以实现当前目标,那么领域专家系统对该目标就失去效用。

在知识不完备的情况下进行决策,大多属于基于概率的决策。E. H. Shortliffe等人最早在决策系统中引入可信度的概念来解决概率决策问题[6]。基于概率的决策方法主要有机器学习方法[7,8]、神经网络方法[9,10]和不确定性推理方法[11]等。这些方法的基本特征是:以实验和统计为理论基础,用大量的重复实验或同质经验(数据)对目标和前提条件进行分析

到稿日期:2011-04-18 返修日期:2011-07-28 本文受国家自然科学基金(60970042)资助。
夏兰亭(1982—),男,博士生,主要研究领域为知识表示与推理;韩道军(1979—),男,博士生,讲师,主要研究领域为软件工程、机器学习、形式概念分析;边 芮(1982—),女,博士,主要研究领域为知识工程与应用、智能规划和调度等;李 磊(1951—),男,教授,主要研究领域计算机软件、数据库与知识库。

统计，以获得实现目标的概率[12]。基于概率的决策方法用实现目标的概率来定义最优解的标准，即在若干个可实现目标的方式中，概率最大者为最优解。在现实中，如果无法对决策问题进行实验和统计，那么基于概率的决策方法就不能给出目标和前提条件的度量，该方法也将失去效用。

本文在多前提决策领域中，以“知识不完备”和“无法实验与统计”为前提研究最优解的求解策略。给定一个待解决的目标和该目标所有实现方式的知识。当某个实现方式不能实现目标时，确定该实现方式中当前不满足的所有前提条件，然后对其可信度进行评估，以获得该方式的实现难度。当所有的实现方式都不能实现目标时，就根据实现难度选择最容易的实现方式。其中前提条件的可信度由其推论与现实情况的符合程度来度量。

本文选择最优解的策略是把实现难度最小的实现方式作为目标最优的实现方式。

本文第 2 节介绍相关的假定，引入目标和条件集的概念；第 3 节给出最简条件集的递归定义，并在此之上提出一个极小最简不完备条件集算法，将目标的前提条件分解成最简单的形式，以便确定当前不满足的部分；第 4 节对条件集的实现难度进行评估，并证明该评估是可计算的；第 5 节提出评价目标前提的逻辑框架 O，并证明该框架的可计算性和推理能力，然后给出框架 O 的实现和应用。

2 基本概念

本节先给出与本文研究内容相关的假定，然后引入软件工程领域的一个实例作为背景，最后给出与条件集相关的概念。

本文是在一阶逻辑中讨论多前提决策的最优解问题，相关的研究是在以下假定中完成的。

(1) M 是一个一阶的公理系统，L 是其语言，$K\cup F$ 是其公理集，IR 是其推理规则集。

· 语言 L 是一阶逻辑语言，包括符号表、项、原子和公式；

· 动作 A 是一个形如“$\alpha\Rightarrow\beta$”的公式，其中 α 和 β 都是合式公式。公式“$\alpha\Rightarrow\beta$”表示存在一个以 α 为前提的动作，其效果使 β 成立。

语言 L 比经典的一阶逻辑语言[13]多了一个逻辑连接符“$\Rightarrow$”和一类表示动作的公式“$\alpha\Rightarrow\beta$”。除此之外，语言 L 与经典一阶逻辑语言的定义相同。

· 知识集 $K=K^a\cup K^d$（$K^a\cap K^d=\emptyset$）是一个有穷的公式集。对于具体的问题领域，K^a 为该领域的动作集，K^d 为除动作以外的领域知识集。

· 事实集 F 是一个有穷的基公式集。

· 式子 $X\vdash\varphi$ 表示公式 φ 可由公理集 X 推出。

(2) 文字是原子或者原子的否定。

(3) 目标是一个基公式。

在系统 M 中，对于任意一个给定目标 G：

(1) 若 $K\cup F\vdash G$，则目标 G 是可由公理集 $K\cup F$ 推理实现的，其推理序列就是现实目标 G 的具体步骤（或实现方式）。这种可推理实现目标的最优解问题是一个经典的研究问题[2,3,6]，本文对此不作进一步的探讨。

(2) 否则，目标 G 是不可推理实现的，即 G 的所有实现方式都不能实现目标。本文通过评估目标 G 所有实现方式的实现难度来确定 G 的最优实现方式。这是本文的研究重点。

在本文的研究中，对任意一个不可推理实现的目标 G，先利用 K^a 得到目标 G 的所有实现方式及每个实现方式的所有前提条件，然后根据 F 确定这些实现方式中当前不满足的前提条件，最后利用 K^d 和 F 评估所有实现方式的实现难度，并由此确定目标 G 的最优实现方式。

下面以软件工程领域中的原材料数据库维护（简称材料数据库领域，记为 m）为背景对相关符号的含义进行说明。

假设制造企业需要维护其原材料的库存信息，包括库存表、库存明细表、货位表、报废表、出仓表、入仓表和调拨表。这些表之间的关系如图 1 所示。

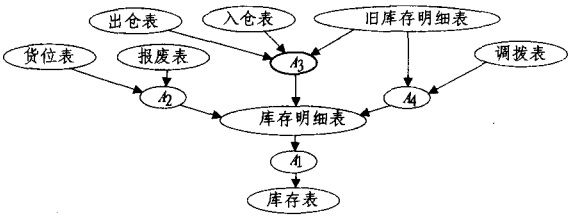

图 1 材料数据库表与动作的关系示意图

(1) 旧库存明细表是指发生数据更新前的库存明细表。

(2) 动作集 $K^a(m)=\{A_1,A_2,A_3,A_4\}$，其中 A_i 以输入箭头的表为其前提，输出箭头的表是其运算或处理后的结果，$i=1\cdots4$。图 1 中动作 A_1，A_2，A_3 和 A_4 可形式化表示为：

A_1：库存明细$(X)\Rightarrow$库存(X)，其以库存明细表为前提，对明细表中相同材料进行汇总产生库存表。

A_2：货位$(X)\wedge$报废$(X)\Rightarrow$库存明细(X)，其以货位表和报废表为前提，在货位表中删除废品所占货位后产生库存明细表。

A_3：出仓$(X)\wedge$入仓$(X)\wedge$旧库存明细$(X)\Rightarrow$库存明细(X)，其以出仓表、入仓表和旧库存明细表为前提，在旧库存明细表的基础上增加入仓材料，删除出仓材料，产生新的库存明细表。

A_4：调拨$(X)\wedge$旧库存明细$(X)\Rightarrow$库存明细(X)，其以调拨表和旧库存明细表为前提，在旧库存明细表的基础上根据调拨表进行材料记录的“增加、删除和修改”操作，产生新的材料库存明细表。

显然，由图 1 可知，一个库存明细表可有 3 种实现动作：A_2、A_3 和 A_4，每个实现动作都有各自的前提条件。

假设当前的目标是利用动作 A_1、A_2、A_3 和 A_4 建立原材料的库存明细表。

(1) 若动作 A_i 的前提条件是完备的，则可直接用动作 A_i 产生出库存明细表，$i=2,3,4$。

(2) 若所有动作的前提条件都不完备，则没有动作可直接产生库存明细表。将对动作 A_2-A_4 中不完备的前提条件进行评估，并根据每个动作的实现难度来确定最优的实现动作。

在研究中发现，任意一个问题领域都存在其特定的领域知识。在求解该领域的目标时，运用其领域知识可提高求解效率和质量。

在材料数据库领域 m 中，除动作之外的领域知识集 $K^d(m)$ 如表 1 所列，该领域中的一个具体实例的事实集 $F(m)$ 如表 2 所列。

表1 $K^d(m)$的公式及其含义

	$K^d(m)$的公式	公式的含义
1	货位(X)→取料(X,货架,N)	如果货位表中有材料X,那么企业会在生产中从"货架"上取料X N表示第N次取料X
2	¬货位(X)→取料(X,W,N)∧W≠货架	如果货位表中没有材料X,那么一定不是在"货架"上取到X N表示第N次从其他位置上取料X
3	出仓(X)→使用(X,N)	如果出仓表中有材料X,那么企业会在生产中使用X N表示第N次使用X
4	¬出仓(X)→使用(Y,N)∧替换(X,Y)	如果出仓表中没有材料X,那么企业会在生产中使用X的替代品Y N表示第N次使用Y
5	入仓(X)→验货(X,N)	如果入仓表中有材料X,那么企业会在某个时候对X验货 N表示第N次验货X
6	¬入仓(X)→采购(X,N)	如果入仓表中没有材料X,那么企业会在某个时候采购X N表示第N次采购X
7	调拨(X)→使用(X,N)	如果调拨表中有材料X,那么企业会在生产中使用X N表示第N次使用X
8	¬调拨(X)→投诉(X,N)	如果调拨表中没有材料X,那么某个部门会投诉没有调拨到X N表示第N次投诉X

表2 $F(m)$的模式和例

F(m)的模式	F(m)的例
库存(X)	库存(铜片)
报废(X)	报废(铜片)
取料(X,W,N)	取料(铜片,货架,1),取料(铜片,货架,2),取料(铜片,货架,3),取料(铜片,货架,4),取料(铜片,_,5),取料(铜片,_,6)
使用(X,N)	使用(铜片,1),使用(铜片,2),使用(铜片,3),使用(铜片,4),使用(铜片,5),使用(铁片,6)
替换(X,Y)	替换(铜片,铁片)
验货(X,N)	验货(铜片,1),验货(铜片,2),验货(铜片,3)
采购(X,N)	采购(铜片,1)
投诉(X,N)	投诉(铜片,1),投诉(铜片,2)

在研究过程中发现,对任意给定的问题领域和该领域的一个实例:

(1) 若知识集$K=K^a\cup K^d$越大,则表示对该领域的理解越清楚,且该领域目标的实现方式也就越多;

(2) 若事实集F越大,则表示对该实例的理解越清楚,且对领域目标实现方式的评估就越准确。

因此,为了获得实现目标的最优方式,通常情况下需要最大限度地收集该领域的知识和该实例的事实。

为了评估实现方式的实现难度,下面给出与前提条件集相关的概念。

定义1 设K是M的知识集,S是一个公式集$\{\alpha_1,\alpha_2,\cdots,\alpha_n\}$,$G$是一个目标。若$K\cup S\vdash G$,则称$S$是$G$在$K$中的条件集,$\alpha_i$为$G$的条件,$i=1\cdots n$。

假设在M中,G是目标,$S_1=\{\alpha_1,\alpha_2,\cdots,\alpha_n\}$,$S_2=\{\beta_1,\beta_2,\cdots,\beta_k\}$,$\cdots$,$S_m=\{\varphi_1,\varphi_2,\cdots,\varphi_t\}$是$G$在$K$中的条件集。

G和条件集$S_1,S_2,\cdots,S_m$之间的层次关系如图2所示。

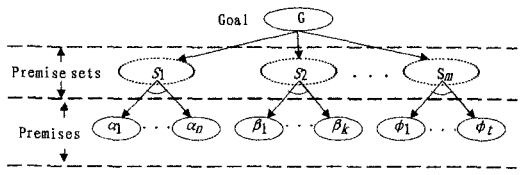

图2 目标和条件集的层次关系示意图

由图2可知:目标和条件集之间的层次关系图是一类"与或"图,即条件集$S_1,S_2,\cdots,S_m$之间是"或"关系,表示目标G的多个实现方式;每个条件集中的条件之间是"与"关系,表示该实现方式的多个前提条件。

由定义1可知,条件集S是G的某个实现方式中所有前提条件的集合。因此,选择目标的最优实现方式就可转化为对实现方式前提条件集的选择。

定义2 假设S是目标G在K中的条件集,F是事实集,

(1) 若$\forall S'\subset S$,都有$K\cup S'\nvdash G$,则称S是G在K中的极小条件集;

(2) 若$\forall\varphi\in S$,或者$K\cup F\vdash\varphi$,或者φ是文字且不存在$\alpha\Rightarrow\beta\in K$使得$K\cup\{\beta\}\vdash\varphi$,则称$S$是$G$在$K$中的最简条件集;

(3) 若$\forall\varphi\in S$,都有$K\cup F\vdash\varphi$,则称S是G在K中的完备条件集,否则称S为不完备条件集。

对于任意一个给定目标G、一个知识集K和一个事实集F,

(1) 若$K\cup F\vdash G$,则G是可推理实现的。由定义2(3)可知,存在G的一个完备条件集S。因此,只需在M中进行相应的推理即可实现目标G。

(2) 否则,G是不可推理实现的。因此,存在G在K中的不完备条件集$S_1,S_2,\cdots,S_m$,当$m>1$时,需要确定实现目标的最优条件集。

下面用材料数据库的例子来说明定义1和定义2中所有概念的含义。

例1 在前面材料数据库m的实例中,$K=K^a(m)\cup K^d(m)$,$F=F(m)$,目标$G=$库存(铜片),且

$S_1=\{$库存(铜片)$\}$,$S_2=\{$库存(铜片),报废(铜片)$\}$

$S_3=\{$库存明细(铜片)$\}$,$S_4=\{$货位(铜片),报废(铜片)$\}$

(1) S_1-S_4都是目标G在K中的条件集。

对于S_1和S_2,有$G\in S_1$,$G\in S_2$,所以,$S_1\vdash G$,$S_2\vdash G$;

对于S_3,由图1可知,用动作A_1可实现目标G,即$\{A_1\}\cup S_3\vdash G$;

对于S_4,由图1可知,须先用动作A_2得到"库存明细(铜片)",然后用A_1实现目标G,即$\{A_1,A_2\}\cup S_4\vdash G$。

(2) S_1,S_3和S_4是目标G在K中的极小条件集,S_2不是目标G在K中的极小条件集。

对于S_1和S_3,由于S_1和S_3只有一个元素,因此它们肯定是G的极小条件集;

对于S_4,由定义1可知,$\forall\alpha\in S_4$,都有$K\cup(S_4-\{\alpha\})\nvdash G$,所以$S_4$是$G$的极小条件集;

对于S_2,因为$S_2-\{$报废(铜片)$\}=S_1$仍然是G的条件集,所以S_2不是G的极小条件集。

(3) S_1,S_2和S_4是目标G在K中的最简条件集,S_3不是最简条件集。

对于S_1和S_2,有$S_1\subseteq F$,$S_2\subseteq F$,所以S_1和S_2中的元素都可由F直接推出,即它们都是G的最简条件集。

对于S_4,因为$F\vdash$报废(铜片),并且"货位(铜片)"是文字并且A_1-A_4的效果都不能推出"货位(铜片)",所以由定义2(3)可知,S_4是G的最简条件集;

对于S_3,虽然$K\cup F\nvdash$库存明细(铜片),但"库存明细(铜片)"可由动作A_2,A_3或A_4的效果推出,由定义2(3)可知S_3不是G的最简条件集。

(4) S_1 和 S_2 是目标 G 在 K 中的完备条件集，S_3 和 S_4 都是目标 G 在 K 中的不完备条件集；

对于 S_1 和 S_2，有 $S_1 \subseteq F, S_2 \subseteq F$，所以 S_1 和 S_2 中的元素都可由 F 直接推出，即它们都是 G 的完备条件集；

对于 S_3，有 $K \cup F \nvdash$ 库存明细(铜片)，因此 S_3 是 G 的不完备条件集。

对于 S_4，有 $K \cup F \nvdash$ 货位(铜片)，因此 S_4 是 G 的不完备条件集。

由分析(1)－(4)可知，S_4 是目标 G 在 K 中的极小条件集、最简条件集和不完备条件集。在本文中，简称 S_4 是目标 G 在 K 中的极小最简不完备条件集。

3 极小最简不完备条件集算法

为确定目标实现方式中当前不满足的前提条件，需要把目标的前提条件分解成最简单的形式(文字)。本节给出最简条件集的递归定义，并在此基础上提出一个极小最简不完备条件集的求解算法，算法可把目标的前提条件分解成最简单的形式。

3.1 最简条件集

下面给出目标最简条件集的递归定义。

定义 3 设 G 是目标，K 是知识集，F 是事实集，依次定义 $S(G)$ 及其递归层 $l(G)$ 如下：

(1) 若 $K \cup F \vdash G$，则 $S(G)=\{G\}, l(G)=1$；

(2) 若所有 $\alpha_i \Rightarrow \beta_i \in K, i=1\cdots k$，且 $K \cup \{\beta_1, \beta_2, \cdots, \beta_k\} \vdash G$，则 $S(G)=\bigcup_{1\leqslant i\leqslant k} S(\alpha_i), l(G)=\max(l(\alpha_i))+1$；

(3) 若存在文字 $\varphi_1, \varphi_2, \cdots, \varphi_k$，使 $K \cup \{\varphi_1, \varphi_2, \cdots, \varphi_k\} \vdash G$ 且 $K \cup \{\varphi_1, \varphi_2, \cdots, \varphi_k\} \nvdash \neg G$，则 $S(G)=\bigcup_{1\leqslant i\leqslant k} S(\varphi_i), l(G)=\max(l(\varphi_i))+1$；

(4) 若 G 是文字，则 $S(G)=\{G\}, l(G)=1$。

下面证明定义 3 中的 $S(G)$ 所具有的性质及其与目标 G 之间的关系。

定理 1 $S(G)$ 是 G 在 K 中的最简条件集。

证明：

(1) 证明 $S(G)$ 是 G 在 K 中的条件集。

下面对 $l(G)$ 用强数学归纳法来证明。

(1.1) $l(G)=1$

由定义 3 可知，$S(G)$ 由递归基(1)或(4)得到，即 $S(G)=\{G\}, S(G) \vdash G$。

由定义 1 可知 $S(G)$ 是 G 的条件集。

(1.2) 假设 $l(G') \leqslant j$ 时，$S(G')$ 是 G' 的条件集。现在证明，当 $l(G)=j+1$ 时，$S(G)$ 是 G 的条件集。

(1.2.1) 当 $S(G)$ 由递归步(2)得到时，$S(G)=\bigcup_{1\leqslant i\leqslant k} S(\alpha_i)$。

由动作 $A: \alpha \Rightarrow \beta$ 的含义可知，α 是使 β 成立的动作前提，即 $\{A\} \cup \{\alpha\} \vdash \beta$。

在递归步(2)中，有 $\alpha_i \Rightarrow \beta_i \in K, i=1\cdots k$。因此，$K \cup \{\alpha_1, \alpha_2, \cdots, \alpha_k\} \vdash \{\beta_1, \beta_2, \cdots, \beta_k\}$。

再由 $K \cup \{\beta_1, \beta_2, \cdots, \beta_k\} \vdash G$ 可得，$K \cup \{\alpha_1, \alpha_2, \cdots, \alpha_k\} \vdash G$。

由定义 3 和 $l(G)=j+1$ 可知，$l(\alpha_i) \leqslant j, i=1\cdots k$。

由归纳假设可知，当 $l(\alpha_i) \leqslant j$ 时，$S(\alpha_i)$ 是 α_i 的条件集，即 $K \cup S(\alpha_i) \vdash \alpha_i, i=1\cdots k$。

因此，$K \cup (\bigcup_{1\leqslant i\leqslant k} S(\alpha_i)) \vdash \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$。

由已证结论“$K \cup \{\alpha_1, \alpha_2, \cdots, \alpha_k\} \vdash G$，可得 $K \cup (\bigcup_{1\leqslant i\leqslant k} S(\alpha_i)) \vdash G$。

由定义 1 可知，$\bigcup_{1\leqslant i\leqslant k} S(\alpha_i)$ 是 G 的条件集，即 $S(G)$ 是 G 的条件集。

(1.2.2) 当 $S(G)$ 由递归步(3)得到时，$S(G)=\bigcup_{1\leqslant i\leqslant k} S(\varphi_i)$。

由定义 3 和 $l(G)=j+1$ 可知，$l(\varphi_i) \leqslant j, i=1\cdots k$。

由归纳假设可知，当 $l(\varphi_i) \leqslant j$ 时，$S(\varphi_i)$ 是 φ_i 的条件集，即 $K \cup S(\varphi_i) \vdash \varphi_i, i=1\cdots k$。

因此，$K \cup (\bigcup_{1\leqslant i\leqslant k} S(\varphi_i)) \vdash \{\varphi_1, \varphi_2, \cdots, \varphi_k\}$。

再由 $K \cup \{\varphi_1, \varphi_2, \cdots, \varphi_k\} \vdash G$ 且 $K \cup \{\varphi_1, \varphi_2, \cdots, \varphi_k\} \nvdash \neg G$ 可得，$K \cup \{\varphi_1, \varphi_2, \cdots, \varphi_k\}$ 是一致的，且 $K \cup (\bigcup_{1\leqslant i\leqslant k} S(\varphi_i)) \vdash G$。

由定义 1 可知，$\bigcup_{1\leqslant i\leqslant k} S(\varphi_i)$ 是 G 的条件集，即 $S(G)$ 是 G 的条件集。

综合情况(1.2.1)和(1.2.2)可知，当 $l(G)=j+1$ 时，$S(G)$ 是 G 的条件集，情况(1.2)成立。

综合情况(1.1)和(1.2)可知，$S(G)$ 是 G 在 K 中的条件集。

(2) 证明 $S(G)$ 是最简的。

令 $S(G)=\{\psi_1, \psi_2, \cdots, \psi_n\}$ 是目标 G 按定义 3 得到的最终结果。

下面用反证法来证明。

假设存在 $\psi_i \in S(G)$ 不满足定义 2(2)，即 $K \cup F \nvdash \psi_i$ 且 ψ_i 不是文字，或者 $K \cup F \nvdash \psi_i$ 且存在 $\alpha \Rightarrow \beta \in K$ 使得 $K \cup \{\beta\} \vdash \psi_i$。

(2.1) $K \cup F \nvdash \psi_i$ 且 ψ_i 不是文字

由于 ψ_i 不是文字，根据公式的构造性定义可知存在文字 $\varphi_1, \varphi_2, \cdots, \varphi_k$ 有 $K \cup \{\varphi_1, \varphi_2, \cdots, \varphi_k\} \vdash \psi_i$。

因此，对 ψ_i 可按定义 3(3)进行递归分解。

这与“$S(G)$ 是目标 G 按定义 3 得到的最终结果”相矛盾。

(2.2) $K \cup F \nvdash \psi_i$ 且存在 $\alpha \Rightarrow \beta \in K$ 使得 $K \cup \{\beta\} \vdash \psi_i$

由于 $\alpha \Rightarrow \beta \in K$ 使得 $K \cup \{\beta\} \vdash \psi_i$，因此 ψ_i 可按定义 3(2)进行分解。

这与“$S(G)$ 是目标 G 按定义 3 得到的最终结果”相矛盾。

综合情况(2.1)和(2.2)可知，无论哪种情况下，都可产生矛盾，因此 $\forall \psi_i \in S(G)$，一定有或者 $K \cup F \vdash \psi_i$，或者 ψ_i 是文字且不存在 $\alpha \Rightarrow \beta \in K$ 使得 $K \cup \{\beta\} \vdash \psi_i$。

由定义 2(2)可知，$S(G)$ 是最简的。

综合情况(1)和(2)，$S(G)$ 是 G 在 K 中的最简条件集。

3.2 算法描述

利用定义 3 和求集合极小元的 Minimal 算法给出目标 G 的极小最简不完备条件集算法 Condition(G, K, F)。算法按照图 2 所示的层次关系输出目标 G 在 K 中的所有极小最简不完备条件集的集合$\{S_1, S_2, \cdots, S_m\}$。

输入：G 是目标，K 是知识集，F 是事实集。

输出：G 的所有极小最简不完备条件集的集合。

BEGIN

(1) If K∪F ⊢G Then Return {{G}};

(2) $GS_i = \emptyset$;

(3) $MK_i = \emptyset$;

For each K′⊆K do

```
    If K∪(∪α⇏β∈K′{β}) ⊢G Then MK:= MK∪{K′};
(4) If MK≠∅ Then {
        MK:=Minimal(MK);
        For each K′∈MK do
            GS:=GS∪{∪α⇏β∈K′S:S∈Condition(α,K,F)};
        Return GS;
    }
(5) PA:= {α,¬α:α∈Atomic(G)};
    MK:= ∅;
    For each A⊆PA do
        If K∪A ⊢G∧K∪A ⊬G Then MK:=MK∪{A};
(6) If MK≠∅ Then {
    MK:=Minimal(MK);
    For each A∈MK do
        GS:= GS∪{∪φ∈AS:S∈Condition(φ,K,F)};
    Return GS;
    }
(7)Return {{G}};
END
```

如果由步骤(1)返回结果，则说明 G 是一个可推理实现的目标。步骤(3)调用原公理系统 M 进行推理，获得了目标 G 所有可能的实现方式。步骤(4)用 Minimal 算法获得目标 G 所有极小的实现方式，对每个实现方式递归地获得其条件集，目标 G 的条件集的集合是其所有实现方式条件集组成的集合。步骤(5)和(6)是相似的求解过程。

下面证明 Condition(G,K,F)算法实现了条件集 $S(G)$ 的递归定义。

定理 2 $\forall S\in$Condition(G,K,F)，若 $S\neq\{G\}$，则 S 是 G 在 K 中的极小最简不完备条件集。

证明：

(1) 证明 S 是最简条件集

Condition(G,K,F)算法的步骤(1)实现了定义 3(1)；

Condition(G,K,F)算法的步骤(3)和(4)实现了定义 3(2)；

Condition(G,K,F)算法的步骤(5)和(6)实现了定义 3(3)；

Condition(G,K,F)算法的步骤(7)实现了定义 3(4)。

由定理 1 可知 $\forall S\in$Condition(G,K,F)都是 G 在 K 中的最简条件集。

(2) 证明 S 是极小的

对于 Condition(G,K,F)算法出口，步骤(1)和(4)的返回值{{G}}只有一个元素，其显然是极小的。

对于 Condition(G,K,F)算法的递归体，步骤(4)和(6)的 Minimal 算法保证了条件集的极小性。

因此，S 是极小的。

(3) 证明如果 $S\neq\{G\}$，则 S 是不完备的

假设 S 是完备的。

由完备条件集的定义 2(3)可知，对 $\forall\varphi\in S$，都有 $K\cup F\vdash\varphi$，即 $K\cup F\vdash S$。

由算法 Condition(G,K,F)可知，其在第(1)步将直接返回条件集的集合{{G}}，所以 $S=\{G\}$。

这显然与已知条件“$S\neq\{G\}$”相矛盾，所以 S 一定是不完备的。

综合情况(1)－(3)可知，若 $S\neq\{G\}$，则 S 是 G 在 K 中的极小最简不完备条件集。

3.3 算法应用

第 3.1 节和 3.2 节分别给出了最简条件集的递归定义和极小最简不完备条件集算法，本节用材料数据库的例子来说明极小最简不完备条件集算法的有效性。

例 2　在材料数据库 m 的实例中，$K=K^a(m)\cup K^d(m)$，$F=F(m)$，目标 G=库存(铜片)，求目标 G 在 K 中的极小最简不完备条件集。

算法 Condition(G,K,F)对“库存(铜片)”的条件集进行递归分解过程如图 3 所示。

由图 3 可知，Condition(库存(铜片)，K,F)的递归分解图是一类“与或”图：条件集之间是“或”关系，表示目标的多个实现方式；每个条件集中的条件之间是“与”关系，表示该实现方式的多个前提条件。其中目标、条件集和条件都是递归定义的。为了图示简洁，第 4 层只给出 S_1 的递归结构，S_2 和 S_3 也都有相似的递归结构。

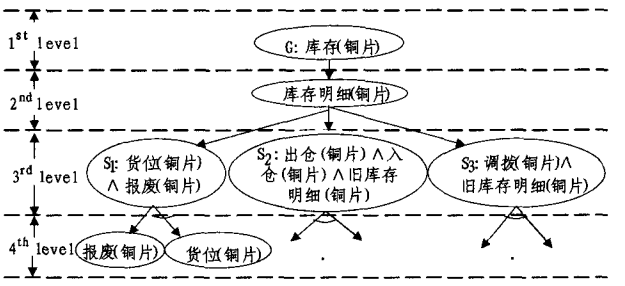

图 3　Condition(库存(铜片)，K,F)的递归求解示意图

在图 3 中，第 2 层的“库存明细(铜片)”是“库存(铜片)”的条件集，第 3 层的 S_1、S_2 和 S_3 是“库存明细(铜片)”的条件集，第 4 层中“报废(铜片)”和“货位(铜片)”是 S_1 的前提条件。显然，{货位(铜片)，报废(铜片)}是“库存(铜片)”的一个极小最简不完备条件集。同理，还可用 S_2 和 S_3 得到“库存(铜片)”其他极小最简不完备条件集。

最终，目标“库存(铜片)”的 3 个极小最简不完备条件集分别是{货位(铜片)，报废(铜片)}、{出仓(铜片)，入仓(铜片)，旧库存明细(铜片)}和{调拨(铜片)，旧库存明细(铜片)}。

例 2 中的目标是建立铜片的库存记录“库存(铜片)”。在图 1 所示领域知识的支持下，其人工求解思路如下：

(1) 如果在事实 F 中找到“库存明细(铜片)”，那么可根据动作 A_1 构造出“库存(铜片)”；

(2) 在事实 F 中没找到“库存明细(铜片)”，则需要利用知识集 K 求“库存明细(铜片)”。

(2.1) 通过知识集 K 的分析可知，A_2 可用来实现“库存明细(铜片)”，因此将求“库存明细(铜片)”转换成求“货位(铜片)∧报废(铜片)”。显然，“货位(铜片)和报废(铜片)”是目标的前提条件；

(2.2) 同理，利用 A_3 可得到“出仓(铜片)，入仓(铜片)和旧库存明细(铜片)”是目标的前提条件；

(2.3) 同理，利用 A_4 可得到“调拨(铜片)和旧库存明细(铜片)”是目标的前提条件。

从这个例子可看出，Condition(库存(铜片)，K,F)的递归求解过程和结果与人工求解的过程和结果相一致。

4 条件集的实现难度

第3节用极小最简不完备条件集算法得到目标所有最简单形式的条件集。本节先对这些条件集中的条件进行评估，以得到它们的可信度，然后综合每个条件集中所有条件的可信度，以得到该条件集的实现难度，最后用材料数据库的例子说明条件集实现难度的计算过程。

下面构造公式 α 与 F 在 K 中相同的逻辑推论集，逻辑推论集将用于条件可信度的计算。

定义 4 设 α 是一个公式，φ 是一个文字，用符号 $C(\alpha,K,F)$ 表示 α 与 F 在 K 中相同的逻辑推论集，且计算公式为 $C(\alpha,K,F)=\{\varphi:K\cup F\vdash\varphi\}\cap\{\varphi:K\cup\{\alpha\}\vdash\varphi\}-\{\varphi:K\vdash\varphi\}$。

定理 3 $C(\alpha,K,F)$ 是能行可计算的。

证明：

1) 证明 $\{\varphi:K\cup F\vdash\varphi\}$ 是能行可计算的

令 $T=K\cup F$。因为 K 和 F 都是有穷公式集，所以 T 也是有穷的。

由假定可知 IR 是 M 的推理规则集。

用 Infer(T) 表示在 T 中使用一次 IR 所能推出的文字集合，即

Infer(T)=$\{\varphi:\varphi$ 是文字 $\wedge\ \varphi$ 可由 $\psi_1,\psi_2,\cdots,\psi_k\in T$ 使用一次 IR 推出$\}$

根据 Infer(T) 的定义构造其算法：

(1) 对 $\forall T'\subseteq T$，用一遍 IR 且 IR 中的每条推理规则至多使用一次的前提下，所得到的结果集记为 $IR(T')$；

(2) 从所有的 $IR(T')$ 中挑选出文字组成集合 Infer(T)。

因为 T 是有穷的，所以求 T 的所有子集和推理规则集 IR 对 T' 进行推理都是能行可计算的。又因为文字的判定是能行可计算的，所以 Infer(T) 是能行可计算的。

下面定义一个迭代算子 $IT(T_i)$：

根据本文对 M 的假定，M 只可能包括有穷个常元和谓词（因为 $K\cup F$ 是有穷的），所以 $K\cup F$ 的推论是有穷的。

$$IT(T_{i+1})=\begin{cases}K\cup F, & i=0\\ IT(T_i)\cup\text{Infer}(T_i), & i>0\end{cases}\qquad(1)$$

由式(1)可知当 $i>0$ 时，$IT(T_i)\subseteq IT(T_{i+1})$，即算子 $IT(T_i)$ 对于 i 单调递增。

所以，存在一个自然数 N，使得 $IT(T_N)=IT(T_{N+1})$。

下面用数学归纳法来来证明当 $IT(T_N)=IT(T_{N+1})$ 时，对 $\forall j>N$，都有 $IT(T_j)=IT(T_N)$。

(1) 当 $j=N+1$ 时

显然，$IT(T_j)=IT(T_N)$，命题成立。

(2) 假设当 $j>N$ 时，有 $IT(T_j)=IT(T_N)$。下面证明 $IT(T_{j+1})=IT(T_N)$。

假设 $IT(T_{j+1})\neq IT(T_N)$。

由归纳假设可知 $IT(T_{j+1})\neq IT(T_j)$。

由式(1)可知 $IT(T_{j+1})-IT(T_j)\neq\varnothing$。

存在一个文字 $\varphi\in IT(T_{j+1})-IT(T_j)$。

由归纳假设可知 $IT(T_j)=IT(T_N)$。

由 Infer(T) 的定义可知，φ 可由 $\psi_1,\psi_2,\cdots,\psi_k\in IT(T_j)$ 使用一次推理规则集 IR 推出。

所以，$\psi_1,\psi_2,\cdots,\psi_k\in IT(T_N)$。

再由式(1)可知，$\varphi\in IT(T_{N+1})-IT(T_N)$，且 φ 由 $\psi_1,\psi_2,\cdots,\psi_k$ 推出。

这显然与已知条件“$IT(T_{N+1})=IT(T_N)$”矛盾。

由归纳证明对 $\forall j>N$，都有 $IT(T_{j+1})=IT(T_N)$。

下面用式(1)来计算算子 $IT(T_i)$：

(1) $i:=0$；

(2) $IT(T_1):=K\cup F$；

(3) $i:=i+1$；

(4) $IT(T_{i+1}):=IT(T_i)\cup\text{Infer}(T_i)$；

(5) 重复步骤(2)－步骤(4)，直到 $IT(T_{i+1})=IT(T_i)$ 为止。

由上面的论证可知，存在自然数 N，使得 $IT(T_N)=IT(T_{N+1})$。

因此，步骤(5)是可终止的，并且 N 是使“$IT(T_{i+1})=IT(T_i)$”成立的最小整数 i。

由已证的结论，当 $IT(T_{N+1})=IT(T_N)$ 时，$\forall j>N$，都有 $IT(T_j)=IT(T_N)$，所以 $IT(T_N)$ 就是 T 能推出的所有文字的集合，即 $IT(T_N)=\{\varphi:K\cup F\vdash\varphi\}$，其中 φ 是文字。

有穷集合的并集是能行可计算的，Infer(T) 也是能行可计算的。由式(1)可知，$IT(T_{i+1})$ 也是能行可计算的。

由于存在自然数 N 使算子 $IT(T_i)$ 可停止，因此 $IT(T_N)$ 是能行可计算的，即 $\{\varphi:K\cup F\vdash\varphi\}$ 是能行可计算的。

2) 证明 $\{\varphi:K\cup\{\alpha\}\vdash\varphi\}$ 和 $\{\varphi:K\vdash\varphi\}$ 也是能行可计算的

证明方法与情况 1) 一致。

综合上述两种情况，由定义 4 可知，$C(\alpha,K,F)$ 是能行可计算的。

对任意给定的一个公式 α，下面研究其在公理系统 M 中的可信度问题。

(1) 若 $K\cup F$ 的逻辑推论是 $K\cup\{\alpha\}$ 的逻辑推论，则表示 $K\cup F$ 支持 α；

(2) 若 $K\cup F$ 的逻辑推论是 $K\cup\{\neg\alpha\}$ 的逻辑推论，则表示 $K\cup F$ 支持 $\neg\alpha$，即 $K\cup F$ 否定 α；

(3) 若 $K\cup F$ 的逻辑推论既不是 $K\cup\{\alpha\}$ 的逻辑推论，也不是 $K\cup\{\neg\alpha\}$ 的逻辑推论，则表示 $K\cup F$ 与 α 无关。

对任意公式 α，推论集 $C(\alpha,K,F)$ 中的元素个数是 $K\cup F$ 支持 α 的个数，$C(\neg\alpha,K,F)$ 中的元素个数是 $K\cup F$ 否定 α 的个数，本文用 $K\cup F$ 支持和否定 α 的个数来反映 α 的可信度。

定义 5 设 α 是一个公式，用符号 $B(\alpha)$ 表示 α 的可信度，

(1) 若 $K\cup F\vdash\alpha$，则 $B(\alpha)=1$；

(2) 若 $K\cup F\nvdash\alpha$ 且 $C(\alpha,K,F)\cup C(\neg\alpha,K,F)=\varnothing$，则 $B(\alpha)=0$；

(3) 若 $K\cup F\nvdash\alpha$ 且 $C(\alpha,K,F)\cup C(\neg\alpha,K,F)\neq\varnothing$，则 $B(\alpha)=|C(\alpha,K,F)|/|C(\alpha,K,F)\cup C(\neg\alpha,K,F)|$。

由定义 5 可知，$0\leqslant B(\alpha)\leqslant 1$。

下面用例 2 中目标 G 在 K 中的极小最简不完备条件集来说明可信度的计算过程。

例 3 在例 2 中，$S_1=$｛货位(铜片)，报废(铜片)｝是目标 $G=$库存(铜片)的一个极小最简不完备条件集，求 S_1 中所有条件的可信度。

(1) $\alpha=$报废(铜片)

由例 2 可知，$\alpha\in F$。根据定义 5(1) 可得，$B(\alpha)=1$。

(2) $\alpha=$货位(铜片)

由定义 4 可知，$C(\alpha,K,F)=\{\varphi:K\cup F\vdash\varphi\}\cap\{\varphi:K\cup\{\alpha\}\vdash\varphi\}-\{\varphi:K\vdash\varphi\}$。

由例 2 中的知识集 K 和事实集 F 可知：

·由 $K\cup F$ 不能推出新的结论可知，$\{\varphi:K\cup F\vdash\varphi\}=F$；

·由领域知识 K^d 可知，$\{$货位$(X)\rightarrow$取料$(X,$货架$,N)\}\cup\{$货位(铜片)$\}\vdash\forall N$ 取料(铜片，货架，N)，即$\{\varphi:K\cup\{$货位(铜片)$\}\vdash\varphi\}=\{\forall N$ 取料(铜片，货架，$N)\}$；

·由 K 不能推出新的结论可知，$\{\varphi:K\vdash\varphi\}=\varnothing$。

所以，

$$C(\alpha,K,F)=\{\varphi:K\cup F\vdash\varphi\}\cap\{\varphi:K\cup\{\alpha\}\vdash\varphi\}-\{\varphi:K\vdash\varphi\}$$
$$=F\cap\{\forall N\text{取料(铜片，货架，}N)\}-\varnothing$$
$$=F\cap\{\forall N\text{取料(铜片，货架，}N)\}$$
$$=\{\text{取料(铜片，货架，1)，取料(铜片，货架，2)，取料(铜片，货架，3)，取料(铜片，货架，4)}\}\quad(\text{表 2})$$

同理，$C(\neg\alpha,K,F)=\{\forall N$ 取料(铜片，$W,N)\wedge W\neq$货架$\}=\{$取料(铜片，_，5)，取料(铜片，_，6)$\}$。

由定义 5(3)可知，$B(\alpha)=|C(\alpha,K,F)|/|C(\alpha,K,F)\cup C(\neg\alpha,K,F)|=4/(4+2)=2/3$。

下面根据上述极小最简不完备条件集的性质来定义条件集的实现难度。

定义 6 设 S 是一个极小最简不完备条件集，用 $D(S)$ 表示 S 的实现难度，计算公式为 $D(S)=\sum_{\alpha\in S}(1-B(\alpha))$。

由定义 5 和定义 6 可知，极小最简不完备条件集 S 实现难度的另一种形式为 $D(S)=\sum_{\alpha\in S}B(\neg\alpha)$。

定理 4 $D(S)$是能行可计算的。

证明：

(1) 证明 $\forall\alpha\in S,B(\alpha)$是能行可计算的

对 $\forall\alpha\in S$，由定理 3 可知，$C(\alpha,K,F)$是能行可计算的，因此 $C(\neg\alpha,K,F)$也是能行可计算的。由定义 5 可知，$B(\alpha)$是能行可计算的。

(2) 证明 $D(S)$是能行可计算的

令 S 是 G 在 K 中的一个极小最简不完备条件集，假设 S 不是有穷的。

由 S 的极小性可知，G 在 K 中的所有条件集都不是有穷集合。但是，$\{G\}$是 G 的一个条件集，并且$\{G\}$是有穷的，这与假设矛盾。因此，S 是有穷的，$E=\{\varphi:\varphi\in S\wedge K\cup F\vdash\varphi\}$也是有穷的。由定义 6 可知，$D(S)$是能行可计算的。

综合(1)，(2)，可知 $D(S)$是能行可计算的。

下面用例 2 中的领域描述和目标的条件集来说明实现难度的计算过程。

例 4 对例 2 中的领域描述和目标 $G=$库存(铜片)，求 G 在 K 中实现难度最小的极小最简不完备条件集。

由例 2 可知目标 G 的 3 个极小最简不完备条件集分别是 $S_1=\{$货位(铜片)，报废(铜片)$\}$、$S_2=\{$出仓(铜片)，入仓(铜片)，旧库存明细(铜片)$\}$和 $S_3=\{$调拨(铜片)，旧库存明细(铜片)$\}$。

(1) 计算 S_1 的实现难度

由例 3 可知 $B($货位(铜片)$)=2/3,B($报废(铜片)$)=1$。

所以，有 $D(S_1)=\sum_{\alpha\in S_1}(1-B(\alpha))=(1-1)+(1-2/3)=1/3$。

(2) 计算 S_2 和 S_3 的实现难度

同理，由定义 5 可知，$B($出仓(铜片)$)=5/6,B($入仓(铜片)$)=3/4,B($旧库存明细(铜片)$)=1,B($调拨(铜片)$)=5/7$。

所以，$D(S_2)=(1-5/6)+(1-3/4)+(1-1)=5/12,D(S_3)=(1-5/7)+(1-1)=2/7$。

(3) 确定实现难度最小的极小最简不完备条件集

由 $D(S_3)\leqslant D(S_1)\leqslant D(S_2)$可知，$S_3$ 是最容易实现目标的条件集。

5 构造评价目标前提的逻辑框架

第 3 节给出了目标的极小最简不完备条件集算法，第 4 节定义了条件集的实现难度。本节将在条件集的实现难度基础上构造评价目标前提的逻辑框架 O，并给出框架的实现原型及应用实例。

5.1 形式化描述

对 M 中的任一目标 G，如果 G 是不可推理实现的，则可使用极小最简不完备条件集算法获得 G 的所有极小最简不完备条件集，然后确定每个极小最简不完备条件集的实现难度。根据综合难度的大小可以得到目标 G 的最优条件集。

定义 7 设 M 是本文假定的一阶公理系统，则评价目标前提的逻辑框架 O 是由以下几部分组成的一阶公理系统：

(1) 以 $L\cup\{C,B,D\}$为语言，其中 C,B 和 D 不在 L 的符号表中；

(2) 以 $K\cup F$ 为公理集；

(3) 以 IR 为推理规则；

(4) 目标 G。

由定义 7 可知逻辑框架 O 是对系统 M 的一个扩充。下面证明框架 O 的推理能力比 M 强。

定理 5 当 $K\cup F\neq\varnothing$ 时，框架 O 的推理能力强于 M。

证明：

由系统 M 的假定可知 $K\cup F$ 是 M 的公理集，IR 是 M 的推理规则集。

(1) 证明 M 所有的定理都是 O 的定理

由公理系统的原理可得，$\{\psi:K\cup F\vdash\psi\}$是系统 M 所有的定理。

在定义 7 中，框架 O 继承了 M 的公理集 $K\cup F$ 和推理规则集 IR。所以，$\{\psi:K\cup F\vdash\psi\}$也是框架 O 的定理。

(2) 证明当 $K\cup F\neq\varnothing$ 时，O 中存在一个定理 ψ,ψ 不是 M 的定理

由 $K\cup F\neq\varnothing$ 可知，$K\cup F\subseteq\{\varphi:K\cup F\vdash\varphi\}$，即$\{\varphi:K\cup F\vdash\varphi\}\neq\varnothing$。

$\exists\alpha\in\{\varphi:K\cup F\vdash\varphi\}$。

令 $\psi=\neg\alpha$。

由“公理系统 M 是一致的”可知，$K\cup F\nvdash\neg\alpha$，即$\neg\alpha$ 不是 M 的定理。

由定义 7 可知，$\psi=\neg\alpha$ 是框架 O 的定理。

所以，O 中存在一个定理 ψ,ψ 不是 M 的定理。

综合情况(1)和(2)，框架 O 的推理能力强于 M。

定理 5 的含义说明如下：

框架 O 继承了一阶公理系统 M 的公理集 $K\cup F$ 和推理规则集 IR，其从公理集 $K\cup F$ 出发进行推理的能力与 M 是相同的。但它比系统 M 多了符号$\{C,B,D\}$，这些新增符号是用来表示 O 中条件集的实现难度。由此可见，框架 O 不仅保留了系统 M 推理的能力，而且增加了对目标的条件集进行评价的能力。即如果目标在 M 中是可实现的，则框架 O 可给出实现步骤(推理过程)，否则其将给出实现目标的最优条件集。

任意给定的目标 G，若 G 在 $K\cup F$ 中是不可推导的，则需执行虚线框中的内容，输出目标 G 的最优条件集。框架 O 的逻辑框架如图 4 所示，虚线框中每个执行步的原理可参考材

料数据库领域相对应的例子。

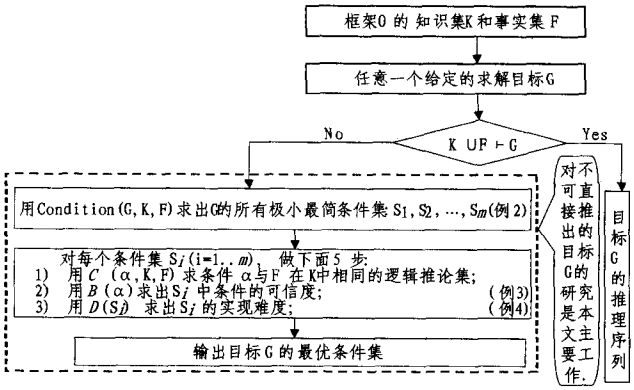

图 4 框架 O 的逻辑框架图

5.2 实现和应用

第 5.1 节完整地定义了评价目标前提的逻辑框架 O，并详细描述了框架 O 对任意给定目标 G 的逻辑处理框架。在此基础上，本节用 Prolog 语言实现框架 O 的原型，并进行系统的原理性实验和实际工程的应用。

逻辑程序理论是一阶理论的子集，因此本文的所有定义和公理在逻辑程序中仍然有效。

用 Prolog 语言实现的逻辑框架具有以下特点：

(1) 具有知识推理的能力。

用 Prolog 语言描述框架 O 中的事实和知识，可得到一个基于知识的领域专家系统。如果目标 G 可由该系统推理得到，那么 Prolog 程序可以输出推理的过程。

(2) 具有评价目标前提的能力。

对于不可推理的目标 G，框架 O 中引入了符号“C,B,D”及其相应的计算公式来分析其最优条件集。Prolog 程序用谓词实现了对应的符号，可对目标 G 的所有条件集进行排序，并输出目标的最优条件集。

(3) 弱化强否定的知识。

框架 O 存在强否定词($\neg\alpha$)，但逻辑程序没有强否定词，Prolog 程序中的知识和事实都没有强否定形式，其否定知识是通过“失败即否定”得到的。

(4) 采用优化技术。

框架 O 在极小最简不完备条件集算法 Condition(G,K,F)中多次调用原系统进行推理，如“$K\cup F\vdash G$”。这种求 $K\cup F$ 所有推论的过程非常耗时，因此在 Prolog 程序中采用 Meta-Programming[14]技术改变 Prolog 的标准编译过程，以优化推理过程。

在用 Prolog 语言实现了框架 O 后，还进行了系统原理性的实验和工程上的应用，以检验其有效性。

(1) 原理性实验

用一个审案专家系统来进行原理性实验，该系统可对任何提问作出回答。当该提问有相应的证据支持时，输出对提问的推理过程。当该提问没有足够证据支持时，系统可根据已有知识和案例输出使该提问成立的最优办案建议，以便辅助办案人员寻找相应的证据。相关研究成果已发表在 ICS-CA2010[15]。

(2) 工程应用

数据的一致性维护是软件工程领域中一项重要的工作。在企业的系统数据库中，原材料、半成品、成品和废料等数据的规模会十分庞大，而且这些数据往往会有多点维护，因此在企业的系统数据库中经常出现数据脏读或数据丢失等一致性问题。

为了解决数据一致性维护问题，运用框架 O 的思想实现了数据库的一致性维护。该系统把数据库中的数据作为事实，把数据之间的关系作为知识。当数据不一致时，可根据已有的事实和知识来确定合理的数据和问题数据。对问题数据，该系统给出相应数据维护的合理建议。在软件所的工程应用，如某钼业集团信息管理系统和市级公共资源交易及电子监察系统等，框架 O 在实际的数据一致性维护中都能给出问题数据的合理建议，收到了良好的应用效果。

结束语 确定目标的最优实现方式是决策领域中重要的研究内容之一。本文针对多前提决策领域中最优问题提出了一种解决方法，即先把目标的前提条件分解成最简单的形式，找出当前不能满足的前提条件，然后给出评估条件集的逻辑框架 O，框架 O 可通过对条件集实现难度的度量确定目标的最优条件集。

智能诊断、机器学习和博弈等领域中也存在多前提决策问题。把本文的研究成果应用到这些领域，以提高相应的求解效率或最佳选择的命中率，将是我们进一步研究的内容。

参 考 文 献

[1] List C, Puppe C. Judgment Aggregation: a Survey[M]. Oxford: Handbook of Rational and Social Choice, 2006

[2] Greco S, Inuiguchi M, Slowinski R. Fuzzy Rough Sets and Multiple-premise Gradual Decision Rules[J]. International Journal of Approximate Reasoning, 2006(41): 179-211

[3] 陈晓红. 决策支持系统理论和应用[M]. 北京：清华大学出版社，2000

[4] Waterman D A. A Guide to Expert Systems [M]. New York: Addison-Wesley, 1986

[5] 欧阳建权，钱跃良. 基于 PDA 的农业专家系统的知识表示与推理策略[J]. 计算机科学，2001，28(11)：40-43

[6] Shortliffe E H. Computer-based Medical Consultations, MYCIN [M]. New York: Elsevier, 1976

[7] Brannon N G, Seiffertt J E, Draelos T J, et al. 2009 Special Issue: Coordinated Machine Learning and Decision Support for Situation Awareness[J]. Neural Networks, 2009(22): 316-325

[8] 程昱，高济，古华茂，等. 基于机器学习的自动协商决策模型[J]. 软件学报，2009，20(8)：2160-2169

[9] Heekeren H R, Marrett S, Ungerleider L G. The neural systems that mediate human perceptual decision making[J]. Nature Reviews Neuroscience, 2008(9): 467- 479

[10] Gold J I, Shadlen M N. The Neural Basis of Decision Making [M]. Annual Reviews, 2007

[11] Fox J, Glasspool D, Grecu D, et al. Argumentation-based Inference and Decision Making—A Medical Perspective[C]// IEEE Intelligent Systems. 2007: 34-41

[12] 周青，彭为. 经典逻辑中的不确定性及其支持度[J]. 计算机学报，2006，29(10)：1882-1888

[13] Enderton H B. A mathematical introduction to logic[M]. New York: Academic Press, 1972

[14] Shapiro E. Concurrent Prolog: A progress report[R]. Fundamentals of Artificial Intelligence, 1986: 277-313

[15] Xia Lan-ting, Gao Jie, Li Lei. Knowledge-based Conditional Reasoning System[C]// Seventh International Conference on Scientific Computing and Applications. 2010: 53-59