

溯因推理研究:现状与问题^{*}陈荣¹ 姜云飞¹ 林笠²(中山大学软件所 广州510275)¹ (暨南大学数学系 广州510632)²

Study of Abductive Reasoning: State of the Art and Problems

CHEN Rong¹ JIANG Yun-Fei¹ LIN Li²(Software Institute, Zhongshan University, Guangzhou 510275)¹ (Mathematics Department, Jinan University, Guangzhou 510623)²

Abstract The notion of abduction was first introduced by the philosopher and logician C. S. Pierce, who claims deduction, abduction and induction are three different forms of human reasoning. Broadly speaking, abduction is a reasoning process invoked to explain a puzzling observation. The study of logic-based abduction has been the most active research in AI, and there are many results around the subject. This paper surveys the state-of-the-art research and presents some problems for future work.

Keywords Abduction, Induction, Logic-based abductive reasoning, Logic programming

在人类认识世界的过程中,现实的生存世界里会不时发生着一些“令人惊奇的”现象,为了解和分析这个问题,人们所使用的解疑释惑的方法往往就是尝试寻找引起这些现象的原因是什么。例如早上醒来你发现路是湿的,你会猜测昨天下雨了。通俗地讲这种解释观测事实(或已知结果)的推理过程就是溯因(Abduction)。可以说类似的例子一直在我们的日常生活或者科学研究中重复着,以现实生活中的医学诊断为例,当医生了解了患者的病症后,他会根据自己对疾病和症状间因果关系方面的医学知识,推断出可能的病因是什么。对于自然科学研究也是如此,N. R. Hanson 和 C. S. Pierce 曾分别论证说,当开普勒断言“火星的运行轨迹是椭圆的”的时候,他所使用的推理方法就是溯因。

最早提出溯因这个概念的人是身兼哲学家和逻辑学家的 C. S. Pierce。不过溯因推理模式的历史可以上溯至中世纪以前,C. S. Pierce 曾经把它和亚里士多德的归纳(Aristotle's Apagoge)做过类比。此后拉普拉斯广泛讨论了从结果到原因的推理,并把它视为自然科学发现的一种重要方法学。众所周知人工智能的一个中心主题就是研究、模拟和实现人类的认知行为,因此溯因推理自然会成为该领域的一项研究内容。1973年,Pople 开创了在人工智能领域研究溯因推理(Abductive Reasoning)的先河。1985年,Charniak 和 McDermott 强调了溯因作为有别于归纳和演绎推理形式的重要性。这直接导致了80年代末期溯因推理真正引起人工智能学者的研究兴趣。此后许多研究发现,溯因可以在人工智能的各个研究领域和各种研究层面上起到独特的作用。

1 关于溯因

在 Pierce 的早期理论中,他提出演绎、归纳和溯因是三种不同的推理模式。可以用三段论的形式阐述这三个概念的不同含义:

演绎:规则——这个袋中的豆子都是白色的,

事例——这些豆子来自这个袋子,

结果——这些豆子是白色的。

归纳:事例——这些豆子来自这个袋子,

结果——这些豆子是白色的,

规则——这个袋中的豆子都是白色的。

溯因:规则——这个袋中的豆子都是白色的,

结果——这些豆子是白色的,

事例——这些豆子来自这个袋子。

据此我们可以发现,演绎是从规则和事例(Case)推知结论的行为,它是唯一一种完全确定的推理;归纳是从结果和事例产生规则的过程;而溯因是在一个已知规则下从一个结果(或者是观察)而认定某事例成立的过程。

溯因和演绎的区别是明显的。溯因是用于产生解释的,而演绎通常被用于检查推论。Pierce 曾把溯因解释成三个步骤:

(1)观察到了一个令人惊讶的事实。

(2)但我们知道,若 A 为真,C 就是无疑的(Matter of Course)。

(3)因此,有理由怀疑 A 是真的。

因此我们可以把溯因推理近似描述成为下面这样一个推理规则:

$$\frac{C \ A \rightarrow C}{A}$$

这条规则很像逆向使用的假言推理(Modus Ponens),然而经典逻辑中假言推理需要的是实质蕴含,而溯因对此没有什么强制性的要求,符号 \rightarrow 只是表达了 A 与 C 之间的关系,对此用户具有更大程度的选择空间,而不必局限于特定的逻辑推理关系、特定类型的观察和解释形式。通常对符号 \rightarrow 的一种自然理解是把它看成一种因果关系,即 $A \rightarrow C$ 表示 A 是 C 成立的原因。由于常识知识是不完全的,因此演绎无法涵盖推理者在常识下做出的所有判断。不过可以把溯因推理看成是一种面向目标的逆向推理过程,因此我们将会看到在背景理论不完全的情况之下如何使用演绎得到解释。溯因自然推广了演绎,因为对于背景理论的那些演绎定理,它们的解释为空集。

表面上看,溯因和归纳的联系更紧密,这两个概念容易变得混淆起来,因为从逻辑的角度看二者的推理方向与标准演

^{*} 本文得到广东省自然科学基金(011162)、中山大学青年教师科研启动基金资助。

绎正好相反,二者的结论都只能在一定信度下予以接受。归纳常常被用于从大量例子中学习的过程,它也要创建理论来解释已知事实。从这个意义上说,溯因似乎是归纳的特例。准确地讲,我们可以把溯因看成从单一观察到它的解释的过程;而归纳是从众多特定实例到一般性结论的枚举归纳(Enumerative Induction)的过程,正因为这样的差异,使得一般情况下归纳假设是谓词子句;而溯因假设往往是命题子句。归纳解释的是一组观察或现象;溯因只解释单个观测结果。归纳需要为进一步的观察做预测;而溯因并不直接对后续的观察负责。虽然在以归纳逻辑程序设计为基础的学习系统中,有效利用背景理论会方便得到问题的固有结构,提高学习质量,但本质上归纳是不需要背景理论的;而溯因需要依赖一个不完全的背景理论来构造并检查所生成的解释。

上面的分析表明,溯因、归纳和演绎构成了人类推理行为的“铁三角”(如图1所示)。

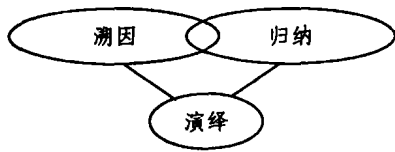


图1 三种推理模式

溯因推理的一个自然应用就是生成因果解释,例如,我们考虑这样一个领域理论:下雨路会湿,草坪也会湿;洒水器打开时草坪也会湿;阳光普照、天气又热时要打开洒水器给草浇水;阳光普照和下雨是不可能同时发生的。我们用图2表示这些常识,



图2 因果理论的简单实例

如果我们观察到“草坪湿了”,在图中所示的常识下我们会得到三个有意义解释,即“下雨”、“阳光普照,天气又热”和“洒水器打开了”,而解释“下雨时洒水器打开了”显得没有意义。从这个实例可以发现,对于一个观测事实,会存在多个可能的解释,这是溯因推理的一个重要特点。不同的解释具有不同的可能性,有时它们在直觉上的简单性也有所不同,比如“阳光普照,天气又热”是“洒水器打开了”的原因,因此比较起来它更基本、更简单。我们还可以理解不同的解释有时会冲突,比如“阳光普照,天气又热”和“下雨”是有矛盾的。因此总体来说溯因是两个相互关联的假设产生和假设选择的过程,选择最佳的解释是溯因推理无法回避的问题之一。

若从逻辑的角度看,溯因推理就是一个找解释的过程,即已知一个背景理论 Σ (公式集合)和一个观察 O (原子公式), O 的解释 Δ 满足:(1) O 是 $\Sigma \cup \Delta$ 的逻辑结果,(2) $\Sigma \cup \Delta$ 是相容的。

2 溯因在逻辑程序设计中的作用

常识推理的一个共同特性是在不完全知识下的缺省推理,非单调逻辑提供了不完全知识下的推理方法,它能在有限知识的基础上得出一些结论,而当知识进一步丰富时这些结

论是可撤销的。在逻辑程序中使用失败即否定规则 NAF (Negation as Failure)可以完成缺省推理,NAF 为非单调的推理提供了一种自然而有力的机制,非单调逻辑中的许多概念在逻辑程序中都有一个自然的对应物。Dix 曾指出非单调推理激发了完美模型的直觉。

为使逻辑程序成为理想的知识表示和推理的实用工具,人们扩展了传统逻辑程序的句法,以包含强否定、析取式的头部等等。已有的研究表明,语法扩展后的逻辑程序具有更强的表达能力,它可以表示行动、时序规划(Temporal Projection)、法律条文和其它形式的缺省信息等多种知识。严格来说一个广义的析取程序就是由形如 $A_1 \vee \dots \vee A_n \leftarrow B_1, \dots, B_m, \sim C_1, \dots, \sim C_k$ 的规则组成的一个有限集合(其中 \sim 代表 NAF, A_i, B_j, C_k 是文字)。当允许 A_i, B_j, C_k 是负文字时,称程序是扩展的; $s=0$ 时称程序是正的; $n=0$ 时称程序是正规的。这样就有了程序的分类。

句法的扩展致使原有语义变得不直观、不完全,甚至于有严重的缺陷。这决定了定义合理语义框架是一个首先要解决的问题,它成为了90年代的一个研究热点。逻辑程序的语义学问题激起了众多人的研究兴趣,自然出现了许多处理方法,但归纳起来看,对于有定程序类,有最小模型 MP;对于分层程序,有 Apt 等人的最小支持模型 $M_{r, \text{min}}$;对于正规逻辑程序类,有 Clark 的完备化 COMP 及 Fitting 对它的三值扩充 COMP₃, 稳定模型 STABLE 及其变种 STABLE' 和 STABLE⁺、良基模型 WFS 及其变种 WFS', WFS⁺, GWFS, 择优模型 PM, 常规语义 REG-SEM, O 语义 O-SEM 等等;对于正析取程序,有广义闭世界假定 GCWA 及其弱化形式 WGCWA;对于分层的析取程序,有完美模型 PERFECT 及其弱化 WPERFECT;对于析取程序类,有析取良基语义 GDWFS, WF³ 和 D-WFS, Przymusiński 的静态语义 STN 和静止语义 Static-SEM 等等;对于扩展程序类,有回答集语义 ANS 以及次协调良基模型 WFSX_r;对于扩展的析取程序,有次协调稳定 STABLE_r、半稳定 Semi-STABLE、回答集 ANS 等语义;而对于广义扩展的析取程序有回答集语义 ANS。虽然人们针对逻辑程序的语义提出了多种方法,但似乎总可以构造实例,使已知方法不能完全抓住程序的理想含义。这表明逻辑程序语义问题没有得到彻底解决,不过现有的这些重要的结果为人们的进一步研究奠定了坚实的基础。

1989年 Eshghi 和 Kowalski 发现 NAF 与溯因这两种机制是相通的,他们论证说“把否定条件看成诱因,再强加析取式作为约束,就可以模拟失败即否定”。这也就是说把含 NAF 的文字看作假设,如果它们在程序的背景之下满足一组一致性约束的话,就假定它们成立。Eshghi 和 Kowalski 的作法是,把正规程序 P 转换成溯因框架 $\langle P^*, A^*, I^* \rangle$, 其中 P^* 是把 P 中所有形如 $\sim p(t)$ 的文字替换成 $p^*(t)$ 而得到的新程序; A^* 是所有形如 $p^*(t)$ 的文字的全体; I^* 是一组形如 $\forall x \rightarrow [p(x) \wedge p^*(x)]$ 和 $\forall x [p(x) \vee p^*(x)]$ 的一致性约束,它们在捕捉 NAF 的含义方面起到了关键的作用。更为重要的是二人为计算溯因解释提供了一种有效的证明过程。“NAF 即溯因”的语义研究方法建立了以逻辑程序为基础的溯因推理框架,同时推广了传统语义研究方面的重要结果,例如广义稳定模型和择优扩充推广了传统意义上的稳定模型。

Eshghi 和 Kowalski 的开创性工作扩展了逻辑程序设计的基础,加深了逻辑程序设计的研究深度,相关的研究工作很多,例如按照时间的顺序排列有代表性的工作,就有文

[EK89],[KakMan90],[Dung91a],[Dung91b],[DenSch92],[SatIwa92],[DenSch93],[Dung93a],[Dung93b],[BTK93],[Brew93],[BrewKon93],[Dung95a],[DamPer95],[DunSun95],[BLMM95],[InoueSak96],[GMS96]以及[InoueSak99a]等等,这些结果构成了逻辑程序说明语义和过程语义研究的一条重要的研究主线,由此开创了一个新的研究课题——溯因逻辑程序设计 ALP (Abductive Logic Programming)。

3 溯因在基于模型的诊断中的作用

近十多年来,以模型为基础的自动诊断的研究工作逐步走向了成熟,其应用也越来越广泛。开创性的工作为自动诊断奠定了坚实的理论基础,同时技术也成熟到了足以解决实际工业问题的程度,例如 Sachenbacher 和 Struss 用诊断系统求解防抱死系统(anti-lock braking system)中的故障,Schroeder 等人的诊断系统 REVISE 成功应用到了公共电话网络中警告关联(Alarm-Correlation)的诊断上。通俗地讲,当观察到某系统的实际测量值不符合该系统的预期行为,即出现了差异(Discrepancy)时,诊断问题就产生了。由于相信系统模型的设计和构造是可靠的,那么差异就是由系统中的局部故障造成的,诊断的任务就是定位这些可能是不可见的故障来源。

基于模型的诊断方法往往使用逻辑公式表示系统的结构、行为描述和观测集。在用逻辑模型化系统的过程中,包含了与一个特殊的谓词集合,用以描述部件工作的状态(正常还是异常),诊断解就用该集合的子集的形式表示出来。基于模型的诊断方法可以大致分成两类,一是基于相容性的诊断(亦称为基于一致性的诊断),二是溯因诊断。最初基于相容性的诊断方法所模型化的是目标系统的正常行为,它所给出的是极小诊断,也就是说它只要求诊断解能够消解差异所隐含的矛盾即可;而溯因诊断更进一步,它还要求观测的某个子集是模型和诊断解的逻辑结果。溯因诊断源自医学诊断问题,最初它所使用的反映疾病与症状的因果关系模型(可以看成故障模型的一个特例),并且模型的知识表示方法与基于相容性的方法有所不同。Konolige 证明了在因果理论中增加封闭公理后,两种方法得到的诊断是一样的。Preist 和 Eshghi 论证说在带有失败即否定的逻辑中基于相容性的诊断与溯因诊断互为对偶。Console 和 Torasso 把基于模型的诊断看成是带一致性约束的溯因诊断,依据这样的观点他们给出了诊断的统一框架,该框架使得诊断的不同逻辑定义有了共同的比较基础。

基于相容性的方法对诊断空间的限制太弱,它得到诊断解要比溯因诊断解多,有些解会是无用的。相比之下溯因诊断的含义更强,它能够使我们集中关注有意义的诊断解。Poole 和 Console 等人讨论了用系统的正常行为模型扩展溯因诊断,使得溯因诊断的适用范围进一步扩大。为了控制诊断过程,有些学者在诊断策略和择优方面做了研究,例如,文[DNPS95],[FNS94]和[NFS95]等。

4 其它应用领域

以常识为基础的缺省推理(Default Reasoning)考虑的是在不引起矛盾的情况下如何使用一般性的规则得出结论。在相关的研究中,最有代表性的非单调逻辑都是带有模态算子或者复杂不动点方程的新逻辑,例如自认知逻辑、缺省理论等。Poole 等人根据溯因提出了一种缺省推理的形式框架理论家(Theorist)。该框架的吸引人之处在于,仍然使用单调逻

辑作为理论的背景描述(可能是不完全的),用满足一定条件的假说来增强背景理论。在推理过程中人会不断尝试使用缺省假设,致使推理呈现出非单调的特点。这就是说,可以使用溯因观点对缺省逻辑的重新阐述。基于这样的认识,Satoh 为缺省逻辑提供了一种“自顶向下的”可靠且完备的证明过程。

当向知识基中增加新知识时(即知识同化,Knowledge Assimilation),一种可能的情况是新知识与原知识基没有冲突。如果能在原知识基中确定新知识成立原因,就可以凝聚新知识基。

溯因在人工智能的各个子领域都可以找到实实在在的应用,例如用溯因解释含混的语句,把规划看成对给定目标状态的解释等等。近年来,无论在深度和广度上,溯因的应用都有了极大的发展,它已经深入到诊断、规划、自然语言理解、知识库更新、信念修正、逻辑程序设计、机器学习等问题领域,起到的作用也越来越成熟。

5 目前研究中求解溯因的方法

5.1 溯因方法的分类

AI 中的溯因推理可分成三种不同的形式:

- (1)集合覆盖模型(Set-covering model, 例如文[ABTJ 87])
- (2)基于逻辑的方法(例如文[EK89])
- (3)知识层次的溯因(Knowledge level abduction, 例如文[Lev89])

方法(1)首先在医学诊断领域出现,它也是最简单的传统方法。它的特点是用一个固定的映射函数表示症状(观察)和疾病(假说)之间的关联,把溯因问题看成一个假说装配问题(Hypotheses Assembly Problem)。这种方法严重依赖于原有的映射函数,实质上该方法没有真正的假说生成过程,它侧重于使用各种措施分出假说的等级。领域知识的轻微变化都会导致求解困难。

方法(2)使用一阶理论来表示领域信息,这样假说与观察之间的关联依赖于“可证”或“演绎”。这种方法侧重于发现好的计算方法和能融入计算过程的选择假说的标准。它与非单调逻辑有着密切的联系,是最活跃的研究领域。基于逻辑的溯因是本文研究的主要内容。

方法(3)是一种元理论的(metatheoretic)方法,该方法的思想是把领域理论表示为一个 agent 的信念。它极大地推广了基于逻辑的方法,例如它允许在建立假说与观察的关联过程中使用部分演绎。

5.2 有代表性的方法简介

5.2.1 真值维护系统 ATMS 是由 de Kleer 于1986年提出的。作为一个问题求解系统的组件,ATMS 的推理机制的原理是计算每个原子(包括 \perp)的标号(label),即它是该原子为真的充分条件,是该原子的解释。

5.2.2 子句维护系统 CMS 为了推广 ATMS,Reiter 和 de Kleer 用命题逻辑的子句集表示背景理论和观察,于1987年提出了子句维护系统 CMS。他们提出本原蕴涵者(本原蕴涵物)的概念,由于它们所具有覆盖性质,它们能紧致刻画解释空间。

5.2.3 线性归结的方法 令 E 是观察 O 的任意一个解释,根据溯因的对偶定义,可以发现 $\rightarrow E$ 具有这样的性质:它是 $\Sigma U \rightarrow O$ 的定理,但不是 Σ 的定理。因此可以使用归结方法发现解释。

(下转第 36 页)

究。

1 研究更高效的算法。虽然现在已有很多优良的挖掘算法,机器性能也越来越好,但对于日益膨胀的数据量需求,算法有时候仍然是瓶颈,像并行算法研究就是一个很有意义的研究课题。

2 序贯模式的挖掘主要是为了应用,解决实际中的问题,如何将发现的序贯模式,序列规则应用到实际领域中,进行交互式的序贯模式挖掘,不断进行基于约束的挖掘,提供额外的控制方法,允许用户说明和使用约束,引导挖掘系统对感兴趣的模式搜索,如何引入用户(专家)的领域知识等等也是值得关注的课题。

3 与其它类挖掘技术的集成,基于不同媒体的挖掘也将是今后的研究热点。

4 目前所讨论的大多基于关系数据库或事务数据库,设计应用于其它类型的数据库(如面向对象数据库,多维数据库,数据仓库等)的序贯模式挖掘方法也将是十分有意义的工作。

5 研究和开发可视化挖掘技术等也将是今后的研究重点。

结束语 本文简单地浏览了序贯模式挖掘的基本概念和挖掘的各个步骤,并简要分析了序贯模式挖掘的几种常见算法。对当前序贯模式挖掘所面临的问题作了简短的说明,展望了未来序贯模式挖掘的研究方向。

参考文献

1 Agrawal R, Srikant R. Mining sequential patterns. In: Proc. 1995

Int. Conf. Data Engineering (ICDE'95), Taipei, Taiwan, Mar. 1995. 3~14

2 Agrawal R, Srikant R. Mining sequential patterns. Generalizations and performance improvements. In: Proc. 5th Int. Conf. Extending Database Technology (EDBT'96), Avignon, France, Mar. 1996. 3~17

3 Pinto H, Han Jiawei, Pei Jian. Multi-dimensional Sequential Pattern Mining

4 Han J, Pei J, Mortazavi-Asl B, Chen Q, Dayal U, Hsu M-C. Freespan: Frequent pattern-projected sequential pattern mining. In: Proc. 2000 Int. Conf. Knowledge

5 Pei J, et al. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: Proc. 2001 Int. Conf. Data Engineering (ICDE'01), Heidelberg, Germany, April 2001. 215~224

6 Garofalkis M, Rastogi R, Shim K. Spirit: Sequential pattern mining with regular expression constraints. In: Proc. 1999 Int. Conf. Very Large Data Bases (VLDB'99), Edinburgh, UK, Sept. 1999. 223~234

7 Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules. In: Proc. of the 20th Int. Conf. on Very Large Database, Santiago, Chile, Sept. 1994

8 Mannila H, Toivonen H, Verkamo A I. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1987, 1: 259~289

9 Han Jiawei, Pei Jian, et al. FreeSpan: frequent pattern-projected sequential pattern mining, ACM Press, New York, NY, USA, Series-Proceeding-Article, ISBN: 1-58113-233-6, 2000. 355~359

10 Joshi M, Karypis G, Kumar V. A Universal Formulation of Sequential Patterns: [Technical Report No. 99-021]. Department of Computer Science, University of Minnesota, 1999

11 Joshi M V, Karypis G, Kumar V. Parallel Algorithms for Mining Sequential Associations: Issues and Challenges: [Technical Report No. 00-002]. Department of Computer science, University of Minnesota, 2000

(上接第25页)

5.2.4 逻辑程序下的溯因 Eshghi 和 Kowalski 的溯因证明过程是一个相互递归调用的两个过程:第一个是被称为溯因演绎,它类似于 SLD 归结,是一个反驳过程,是反向的、自顶向下的;第二个过程是相容演绎,它增量式地检查可满足性的过程,是正向的、自底向上的。两个过程互为对偶。溯因证明过程是不完备的,它不能求全解。

5.3 溯因任务的复杂性

Bylander 把溯因问题刻画成观察和解释间的函数,证明溯因是 NP-完全问题。Selman 和 Levesque 证明生成全部的解是 NP 难的,他们定义了“支持的选择任务”(Support Selection Task)的概念,并认为它是引起难解性的根本原因;不过他们指出,如果不是生成全部的解,而是找到一个解释的话,是可以发现改进算法的。

6 目前研究中存在的问题

以逻辑为基础的溯因研究是一个十分活跃的研究领域,出现了丰富多彩的结果,但是按照上面谈到的思想,它们还存在不足和缺陷,例如假设选择是溯因推理必须面对的问题。

目前溯因问题中的诱因集合往往是基原子集合,把它扩展为子句或者溯因谓词会使溯因更加灵活。与此相关的研究领域是归纳逻辑程序设计 ILP。ILP 建立了机器学习的逻辑程序框架,它在背景理论下可以生成规则,相关的研究成为了一个热点。在第 1.2 节我们阐述了溯因与归纳的区别和联系,事实上 ILP 中的先验必要性、后验可满足性和后验充分性在溯因的定义中都都有所体现,因此二者的结合会加深溯因研究

的深度,使得基于逻辑的溯因更加符合 Pierce 观念中的溯因。

Kowalski 曾论证说,“为了使逻辑程序设计更适合于人工智能领域的知识表示和推理,需要扩展它包含一些附加的特性,比如真否定、溯因、元级推理、术语推理和约束求解”。这方面的研究正在不断地深入, Maim 指出约束程序设计是溯因的一个特例,因此溯因可以把约束程序的语义统一在一个框架中, Kakas 和 Michael 就做了这方面的尝试。

简而言之,目前研究中仍然存在一些没有或者尚未解决的问题:

(1) 假设选择问题。

(2) ALP 研究中没有阐明一致性约束在构造攻击时的作用。

(3) 逻辑程序的语义需要进一步推广和简化。

(4) 研究和简化辩证理论框架。

(5) 基于相容性的诊断方法给出的诊断空间庞大,其中包含无意义的诊断。

(6) 研究或者扩展溯因算法,甚至于在证明过程中允许更灵活的诱因形式。

(7) ALP 与 ILP 的结合。

(8) 用溯因统一约束逻辑程序的语义。

致谢 吉林大学孙吉贵教授对我们的研究工作给予了很多指导,在此表示感谢。

注:由于篇幅所限,参考文献又较多,敬请读者参见:陈荣,《基于辩证语义的溯因推理及诊断研究》,吉林大学博士论文,2000。

