

综述:知识系统的 V&V 技术*)

曾庆田 梁永全 段 华

(山东科技大学信息科学与工程学院 青岛 266510)

摘要 知识库的异常是影响整个知识系统性能的重要因素之一,因此必须对获取的知识进行校验。本文综述了知识库异常检测和验证的相关研究,给出了异常知识的分类及其危害性,分析了知识库验证困难的原因,介绍了用于知识库验证的静态和动态方法,列举了国际上几个著名的知识库验证工具,并对知识库验证的研究进行了展望。

关键词 知识系统,知识异常,知识表示,知识验证

A Survey on Verification and Validation of Knowledge-based Systems

ZENG Qing-Tian LIANG Yong-Quan DUAN Hua

(College of Information Science and Technology, Shandong University of Science and Technology, Qingdao 266510)

Abstract Knowledge abnormalities including inconsistency, redundancy and incompleteness are the main reasons to affect the knowledge-based system efficiencies, so the knowledge verification and validation (V&V) are necessary. The related works about knowledge V&V are summarized in this paper. The knowledge abnormalities are classified and their harms to knowledge-based system are discussed. However, the knowledge V&V are usually difficult since knowledge representation, update and scale. The static and dynamic methods for knowledge V&V are listed respectively, and some known tools for knowledge V&V are compared. Finally, the future research for knowledge V&V is discussed.

Keywords Knowledge-based system, Knowledge abnormalities, Knowledge representation, Knowledge verification, Knowledge validation

1 引言

知识工程在许多领域得到了广泛的应用,但不同开发工具和应用平台使系统的通信、共享、重用性等性能受到了限制,新的知识系统不能有效利用现有的知识系统,不断开始“重新设计”,造成人力、物力、才力的大量浪费。20世纪80年代中期,世界上出现了知识共享(Knowledge Sharing)的研究热潮^[1~3]。这一研究持续到了20世纪末期,到目前势头有增无减。之所以如此,根本原因就是人们已经越来越认识到两点:第一,计算机能力的进一步提高不能完全依赖于它的运算速度;对复杂的问题,计算机需要大量的与问题背景相关的知识。第二,知识的价值体现在它的可共享性上,计算机所获得的知识如果不能被别的系统或个人所共享,那么它的价值就不能充分发挥出来。与知识共享相关的一个重要问题就是已有知识的获取问题。自从20世纪60年代以来,知识获取成为了计算机研究领域的一个难题。但是,人们的研究重点是开发有效的知识获取方法和技术,能让计算机获取领域专

家的经验,用于一些专家系统,使得专家系统具有专家的问题求解能力。

随着专家系统的不断开发,其实践和技术已变得日益完善和成熟。其中的知识获取工作是一项较为困难而又不可缺少的环节,是专家和知识工程师们不能回避的关键问题。知识获取所引起的知识库不一致性是影响整个系统性能的重要因素之一。在知识获取的过程中,不可避免地存在一些错误的知识或矛盾的知识,常见的原因包括:(1)知识源错误:由于我们获取的知识是来源于文本的,所在知识源本身存在难以避免的疏漏和差错。(2)知识获取时无心出错:知识工程师在知识的规范化表示时,由于失误或是其他疏忽容易产生错误。(3)知识描述不准确:在文本中有大量的知识由于自然语言表达的随意性和含糊性而导致语义的多样性,而知识工程师往往根据自己的理解来表示知识,因此造成同一条知识被解释为不同的语义的情况。(4)知识表示错误:知识工程师在进行知识表示时,由于采用了不恰当表示方法造成错误,或存在对相同知识的前后表示不一致。(5)知识推理的非单调性:由

*)本文工作得到自然科学基金的资助(#60073017 和 #60273019)和科技部重大基金项目基金(#2001CCA03000 和 #2002DEA30036)的资助。

发现、调用和组合的一个全新的解决方案。

参 考 文 献

- 1 李安渝. Web Services 技术与实现. 北京:国防工业出版社,2003
- 2 DAML-S 0.9 Draft Release (2003). DAML Services Coalition. <http://www.daml.org/Services/daml-s/0.9/>
- 3 Fensel D, Bussler C. The Web Services Modeling Framework WSMF. http://informatik.uibk.ac.at/users/c70385/wesm/wsmf_bis2002.pdf. 2002
- 4 Motta E, Domingue J, Cabral L, et al. IRS-II: A Framework and Infrastructure for Semantic Web Services. In: ISWC 2003. LNCS 2870. 306~318
- 5 Fensel D, Benjamins V R, Decker S, et al. The Component Model of UPML in a Nutshell. WWW Proceedings of the 1st Working IFIP Conference on Software Architectures (WICSA1), San An-

tonio, Texas, USA, February 1999

- 6 Clancey W J. Heuristic Classification. Artificial Intelligence, 1985, 27: 289~350
- 7 Fensel D, Motta E. Structured Development of Problem Solving Methods. IEEE Transactions on Knowledge and Data Engineering, 2001, 13(6): 913~932
- 8 Motta E. Reusable Components for Knowledge Modelling. Amsterdam, The Netherlands: IOS Press, 1999
- 9 Sycara K, Lu J, Klusch M, et al. Matchmaking among Heterogeneous Agents on the Internet. In: Proceedings of the 1999 AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford University, USA, 22~24 March 1999
- 10 <http://kmi.open.ac.uk/projects/irs/glossary.html>
- 11 Zave P, Jackson M. Four Dark Corners of Requirements Engineering. ACM. 1996

于知识推理本身可能是非单调的,因此推理出来的知识可能出现矛盾。

不管是哪方面的原因造成的知识错误,都会对知识库造成极大的危害,因此必须采取有效的方法和措施,有效地保证所获得的知识是正确的。本文综述了知识系统异常检测和验证的相关研究,给出了异常知识的分类及其危害性,从多个方面分析了知识系统验证困难的原因,介绍了用于知识系统验证的静态和动态方法,列举了国际上几个著名的知识系统验证工具,并对知识系统验证需要进一步研究的内容进行了展望。

2 知识库异常及其检测必要性

2.1 知识库的组成

知识库是由事实和规则组成的,下面进行一些简单的描述。

设 KB 为知识库,则 $KB=RB \cup DB$, 其中 $RB=\{R_1, R_2, \dots, R_n\}$ 为规则集; $DB=GB \cup IB \cup CB$ 为事实集合, 其中 GB 为目标集, 即知识库的输出; IB 为输入集, 即为了得到推理结论所要给定的事实集合; CB 是语义约束集。

为了方便问题的说明,我们将 KB 中的知识分为以下两类:

定义 1 在 KB 中,问题求解领域中必需的,对问题的描述正确,且不影响 KB 的一致性和完备性的知识称为正常知识。

定义 2 KB 中至少具有以下三个特点之一的知识称为异常知识,(1)问题求解领域中用不到的;(2)对问题的描述不正确;(3)影响知识库的一致性和完备性。

知识库中的异常知识来源于很多方面,例如,获取知识和知识转化过程中的粗心大意,知识获取不全面或表达不全面,对知识库的更新和修改等等,均有可能导致异常知识的产生。

2.2 异常知识的分类

知识库中的异常知识主要包括:冗余、矛盾、循环以及知识库的不完备等。知识获取过程中出现的语法错误在知识编译过程中可被检查处理,在此就不讨论了。

由于知识表示方法的多样性,我们没法针对每一种表示方法都给出知识异常的表现形式。产生式是最常用的知识表示方法,下面我们以产生式规则系统为例,给出知识异常的分类以及具体表现形式。

产生式规则可抽象地描述为包含两个元素的对偶,即“前提-结论”或“条件-动作”对^[4]。假定产生式规则有如下的形式:

IF $\langle \text{条件 } 1 \rangle \text{ AND } \langle \text{条件 } 2 \rangle \text{ AND } \dots \text{ AND } \langle \text{条件 } m \rangle$
THEN $\langle \text{动作 } 1 \rangle \text{ AND } \langle \text{动作 } 2 \rangle \text{ AND } \dots \text{ AND } \langle \text{动作 } n \rangle$

将其表示成一阶谓词的形式为:

$AC_1 \wedge AC_2 \wedge \dots \wedge AC_m \rightarrow CC_1 \wedge CC_2 \wedge \dots \wedge CC_n$

其中, $AC_1 \wedge AC_2 \wedge \dots \wedge AC_m$ 称为规则的前件, $AC_i (1 \leq i \leq m)$ 分别称为前提 i ;

$CC_1 \wedge CC_2 \wedge \dots \wedge CC_n$ 称为规则的后件, $CC_j (1 \leq j \leq n)$ 分别称为结论 j 。前提的形式通常为:

$\langle \text{对象} \rangle \langle \text{属性} \rangle \langle \text{操作符} \rangle \langle \text{值} \rangle$

其中,“ $\langle \text{对象} \rangle$ 、 $\langle \text{属性} \rangle$ 、 $\langle \text{值} \rangle$ ”组成了一个“对象-属性-值”(简记为:“O-V-A”)的三元组, $\langle \text{操作符} \rangle$ 的从集合 {是, 不是, =, \neq , \leq , \geq , $>$, $<$ } 中取值。

在产生式系统中,知识库中异常知识的分类如表 1 所示。

表 1 知识库中异常知识的分类

知识库中异常知识的分类			
冗余	矛盾	循环	完备性
<ul style="list-style-type: none"> ● 等价规则 ● 从属规则 <ul style="list-style-type: none"> ■ 条件从属 ■ 结论从属 ● 传递冗余 	<ul style="list-style-type: none"> ● 自相矛盾 ● 传递自相矛盾 ● 隐含矛盾 ● 相互矛盾 ● 传递矛盾 	<ul style="list-style-type: none"> ● 规则前件和后件引入同一谓词 ● 规则链中自我引用 ● 隐含循环 	<ul style="list-style-type: none"> ● 多余输入 ● 死节点 ● 不可达目标 ● 未用的属性值 ● 遗漏规则

下面我们给出产生式规则系统中各种知识异常的具体表现形式,如表 2 所示。

表 2 产生式系统中知识异常的具体表现

分类	表现形式
冗余	等价规则 一规则的前件和后件与另一规则的前件和后件完全等价。
	从属规则 两条规则 R1 和 R2 后件相同,但 R1 的前件部分约束比 R2 多,则 R1 是 R2 的从属规则。
	规则 两条规则 R1 和 R2 前件相同,但 R1 的后件部分约束比 R2 多,则 R2 是 R1 的从属规则。
传递冗余	若两条规则链中第一条规则前件相同,而最后一条规则后件是等价的,则称这两条规则链是传递冗余的。
矛盾	自相矛盾 从某一前提直接推得与之相反的结论,称为自相矛盾。
	传递自相矛盾 从某一前提 P(x) 经过一推理链推出与之相反的结论 $\neg P(x)$ 。
	隐含矛盾 在 KB 中存在一推理链,从推理链后面某一节点中能推出与前面节点相反的结论,但这一规则未包含在 KB 中。例如: $P(x) \rightarrow Q(x) \rightarrow \dots \rightarrow \neg R(x)$ 且规则 $R(x) \rightarrow \neg P(x)$ 应当包含在 KB 中,但未显式地包含在 KB 中。
	相互矛盾 $P(x) \rightarrow Q(x)$, $P(x) \rightarrow R(x)$, 且 $Q(x) \rightarrow \neg R(x)$ 应当包含在 KB 中,但是漏掉了。
	传递矛盾 两条规则有相同的前件,但后件中存在相互矛盾的结论。
循环	传递矛盾 若两条规则链中第一条规则前件相同,而最后一条规则的后件中存在相反的结论,则称 KB 中存在传递矛盾。
	在规则的前件和后件引入同一谓词 $P(x) \wedge Q(x) \rightarrow P(x)$, 或 $P(x) \rightarrow P(A)$, 或 $P(x) \rightarrow P(x) \wedge Q(x)$
	规则链中自我引用 $P(x) \rightarrow Q(x) \rightarrow \dots \rightarrow P(x)$
完备性	隐含循环 $P(x) \rightarrow Q(x) \rightarrow \dots \rightarrow R(x)$ 且 $R(x) \rightarrow P(x)$ 应当包含在 KB 中,却未显式地包含在 KB 中。
	多余输入 系统向用户询问 P(A), 但 P(A) 未在任何规则的前提中出现。
	死节点 在 KB 中存在规则 $R_1: P(x) \rightarrow Q(x)$, 并且 $Q(x)$ 不是一个目标,且不在前提集中出现。
	不可达目标 目标 R(x) 不在结论集中,或者在知识库中存在规则 $R_1: Q(x) \rightarrow R(x)$ 且 $Q(x)$ 是不可达的。
	未用的属性值 O-A-V 三元组中,值全部或部分地未被任何规则引用。
遗漏规则	应该存在于知识库中,但尚未包含在 KB 中的规则,称为遗漏规则。

2.3 异常知识对知识系统性能的影响

知识库是知识系统的核心组成部分,知识库的异常对知识系统存在着不可低估的影响,具体表现在:

(1)知识冗余会影响系统的推理效率,对系统的结论影响甚小。乍看起来冗余对知识系统的影响是无害的,其实则不然。首先,冗余的存在增加了知识库维护的难度。另外,冗余往往是知识编码过程中的错误。在用不确定性进行推理的系统中,冗余可能使得某些事实和结论的可信度改变,以至于影响到系统最后的选择。

(2)知识矛盾对系统性能的影响比知识冗余大。当知识库中含有自相矛盾的规则时,事实集中的某些事实将被删除(因为它们是相互矛盾的,相当于不知道这些事实)。当知识库中的规则通过不同的推理路径得到相反的结论时,系统所给定的结论取决于推理机的选择策略,错误的结论可能导致错误的动作,有时,这种错误动作对系统来说可能是致命的。

(3)知识循环对用反向推理机的系统的影响更大,因为反

向推理是目标驱动的,若目标在一循环链中,经过一个循环,目标本身将成为某一子目标的子目标,这样就导致一系列子目标的无限循环。在正向推理中,循环会影响系统的效率。在利用不确定性进行推理的系统中,事实、中间结果或目标的可信度在循环过程中可能被错误地增大减小。

(4)知识的不完备性对系统得到的结论有很大的影响。它们可以影响到系统的效率。系统可能得到错误的结论,或者在某些情况下,得不到任何结论,这是知识系统用户不希望看到的。在使用过程中遇到这种情况可能导致使用者对知识系统丧失信心。

2.4 知识系统的 V&V 技术

如前节所述,冗余、矛盾、循环、不完备性对知识系统的影响很大,在知识库的建立和知识系统开发过程中应该予以克服,这就需要知识库的 V&V 技术,V&V 是英文 Verification 和 Validation 的缩写。根据 IEEE 术语解释^[5],Verification 和 Validation 的含义分别是:

术语	解释
Verification	(1)对系统或者组成部分进行评价,判断给定阶段的结果是否满足这个阶段开始时输入的条件;(2)对程序正确性的形式化证明。
Validation	在研发的过程中或研发结束时,对系统或者组成部分进行评价,判断系统或组成是否满足指定的需求。
V&V	包括 3 个方面的含义:(1)判断系统或组成的需求是否完备和正确;(2)每个研发阶段的产品是否满足阶段开始前的需求和条件;(3)最后的系统或组成部分是否满足具体的需求。

Verification 确保建立的系统是正确,而 Validation 确保建立正确的系统。知识库的 V&V 技术就是对知识库进行检查和验证,发现其中的异常情况,并采取相应的方法和策略予以纠正,以保证整个知识库的完备和准确。

3 知识系统 V&V 的难点

知识自身的许多特点决定了知识库的 V&V 并不是一件简单的事情,如知识的复杂性,需要不同的知识表示方法的综合;知识不断更新,新知识层出不穷,知识库的规模会越来越多;在知识的获取过程中,领域专家的参与是必需的事情。

3.1 知识表示方法的多样性

一般认为,知识表示在知识处理中处于中心地位^[2,3]。因为一方面获取的知识必须表示成某种形式,否则就无法记录知识;另一方面显然没有表示也就根本谈不上运用知识。不但能否把知识表示出来对知识的处理至关重要,而且知识表示的好坏对知识处理的效率和应用范围影响很大,对知识获取和学习机制的研究也有直接的影响。所以,知识表示问题向来就是知识处理中最热门的研究课题之一。目前已经提出了很多知识表示方法,常用的知识表示方法有一阶谓词逻辑、特征表、规则、框架、过程表示、脚本表示、面向对象技术、语义网络等。在更多的情况下,单一的表示方法不能满足知识表示的要求,这就需要几种表示方法的混合。如:国家知识基础设施(简称:NKI)中知识的表示用到了本体、框架、逻辑等多种知识表示方法^[6,7]。陆汝铃院士研制的盘古常识知识库系统中用到了 Agent、本体、语义网络、逻辑程序等多种方法^[8]。

由于多种知识表示方法的混合,往往会引起知识库的异常现象,如规则和面向对象技术的混合。由于对象的分层和属性的继承关系,往往会引起多余的子包含关系和语义冲突的情况。单就知识的一致性而言,由于知识表示方法的多样性,知识一致性检查十分困难。一个不一致的知识库会带来

矛盾的结果,从而不能有效地使用。但是真正严格的一致性判断只能靠谓词逻辑。谓词逻辑在严格性上是没有问题的,但是表示知识不一定方便。可以说没有一个知识库完全是用谓词逻辑的。要判断这些知识的一致性只有把它们转换成谓词逻辑后再进行检查。但是并不是所有的知识表示方法都可以转换成谓词逻辑。有些要转换成模态逻辑、有些要转换成模糊逻辑等等。这是不足为奇的,因为大量的知识都是使用自然语言表述的,而自然语言的完全形式化是做不到的,用完了所有的逻辑也还不够^[3]。而且我们知道,除谓词逻辑外其它逻辑的一致性判断没有成熟的理论。其次,即使都能转换成谓词逻辑,也只能做形式上的一致性检查,难以恰当处理自然语言的丰富语义,因为这种语义是干巴巴的谓词演算难以表达的。还有一点,自然语言的命题转换成谓词逻辑的方式不是唯一的。例如:命题“雪是白的”和“雪的颜色是白的”在语义上等价。因此,它既可以转换成“白(雪)”,也可以转换成“颜色(雪,白)”。但是,“白(雪)”和“颜色(雪,黑)”两个命题从语法上说不构成矛盾。由于这个原因,本来是不一致的内容转换后也不一定检查出来。对于多知识表示来说,与谓词逻辑的转换十分复杂。因此,知识库的一致性判断从理论上来说是不可行的^[3]。

3.2 知识自身的特点与知识库的规模

各个学科都在不断发展,新的知识不断出现,同时有些知识被修正甚至被否定。如何将新的知识加入到计算机已有的知识中是一个重要的问题。每个领域都是不断发展和更新的,这就造成知识库规模越来越大,包含的知识越来越复杂,势必造成 V&V 的难度。如: XCON 这个系统,1979 年才有 200 条规则,到 1989 年,10 年的时间就变成了 17000 条。在国家知识继承设施(NKI)中,每个学科都在不断地发展,各个学科的新知识层出不穷,NKI 这样一个多学科大规模的知识库的 V&V 就更困难了^[6,7,9,10]。

文[3]指出,“对于知识的一致性检查,除了理论上的问题

以外,大规模知识库的检查还是一个实际问题:彻底的一致性检查总是可行的吗?如果在我们的知识库越来越大时,仍坚持每添加一条知识就检查一次整个知识库,从运行效率的角度来看,这种做法是可操作的吗?”对于这些问题,Lenat 是通过把知识组织成一系列所谓的微理论来解决的。它只管每个微理论内部的一致性。至于知识的全局一致性不予考虑。这是一种很现实的态度和方法。如,在盘古常识库中,先把知识进行模块化,Agent、本体、语义网络、逻辑程序等都是知识模块,然后考虑知识模块的一致性。

从知识工程的实践来看,还有另外一种一致性概念,即知识库的相对一致性,有时也称开放一致性。这种一致性往往随着知识的不断加入,知识库的一致性程度也不断地变化。如:一个医学专家系统的知识库,如果对于一组已知的病例都能给出正确的答案,那么这个知识库对于这组病例来说就是正确的。至于其它的病例,则要通过获取这些病例的知识以后才能知道。

至于知识的完备性问题,文[3]指出知识的完备性概念是相对的。所谓完备,有相对于形式系统的完备,也有相对于人的知识的完备。对于形式系统而言,一个理论的完备是相对于一组公理而言的。但是知识库使用者却要求知识库相对于人类常识是完备的。这就提出了一个实现完备性的可行性问题。我们能够把人类的所有知识,包括常识,都封装在一个知识库里面吗?显然不可能。由于做不到这一点,我们只能退而求其次,对完备性加上某种限制。例如:相当于特定人群的知识完备,相对于某个特定领域的完备,相对于某种特殊应用背景的完备,等等。然而,在实践中,连这些要求也往往难以做到。

3.3 领域专家在知识系统中的作用

领域专家对知识系统建设的作用是不可替代的,从知识获取到知识的分析和验证,都需要领域专家的参与,文[11]对面向专家的知识获取进行了专门的研究,包括面向专家的知识获取的问题以及相应的解决策略,面向专家知识获取中双向了解以及知识工程师如何从领域专家那里获取领域知识等等。

但是由于下面的原因,领域专家参与知识库的建立也给知识库的 V&V 造成了很大的困难:

- (1)对于领域专家,好多知识是经验性的,好多都是只可意会不可言传的。
- (2)由于知识库的规模问题,验证的结果正确与否,领域专家在很多情况下很难判断;
- (3)对于同一问题,不同的专家可能有不同的见解,他们之间也很难一致;
- (4)专家投入的精力和费用也是一个很大的问题。

3.4 知识库异常的界定

关于知识异常目前没有统一的标准定义,大多数定义都是以例子的形式给出的。不同的知识表示方法,对知识异常有着不同的理解和含义,关于知识异常分析和验证的不同技术也采用各自的标准。

目前大多数系统都是关注知识库的正确性方面,在知识的可靠性(reliability),易维护性(maintainability),可重用性(reusability),易懂性(understandability)等方面做的工作很少,更没有统一的标准。

4 知识系统 V&V 的方法

虽然,知识库的 V&V 是很困难的,但是由于知识 V&V

的必要性和重要性,国内外的许多结构和研究人员在这方面做了许多卓有成效的工作^[12~14]。

4.1 知识系统 V&V 的流程

具体来说,知识系统的异常检测并修正错误的步骤如图 1 所示。

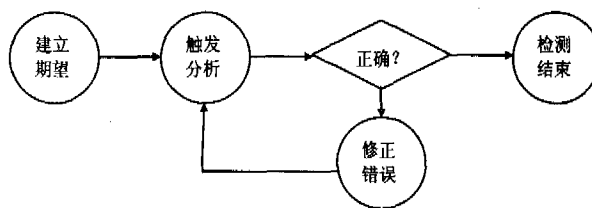


图 1 知识系统检测并修正错误的步骤

(1)建立期望:构造某个行动的前提条件(背景)及相应的结果。

(2)触发分析:当实际事件违反期望时,对照观察数据,重新分析推理过程。

(3)确定错误:通过某些具体方法(以下讨论),发现错误知识的位置和类型。并相应提出修正措施。大体上说,错误类型主要有规则本身有错(前提条件、结论部分或先验概率不准确);规则间的不一致和系统的不完全性 3 种。

(4)修正错误:根据以上确定错误位置和类型,提出相应的修正方法。

(5)重新执行和测试:把修改过的知识重新并入系统,再次重新验证,直至正确为止。

4.2 知识系统 V&V 的方法

图 2 给出了常用的知识分析和验证方法的分类。知识系统 V&V 的方法包括:静态方法和动态方法两种。

(1)静态的 V&V 方法:目标通常是检测完备性和一致性,或者证明程序的正确性。常用的方法有非形式化、半形式化和形式化 3 种。非形式化又包括:读、观察、预排除法等;形式化:通过数学的方法给出严格的数学证明过程。半形式化是介入非形式化和形式化之间的一种方法。

(2)动态的 V&V 方法:主要包括结构测试和功能测试两类方法。结构测试是利用白盒法,测试用例的选择是基于系统执行的细节,包括控制流、数据流、语言细节和程序风格等。功能测试是黑盒法。主要是对系统功能的检测,不太关注具体的细节,因为功能测试关注的焦点是需求描述。

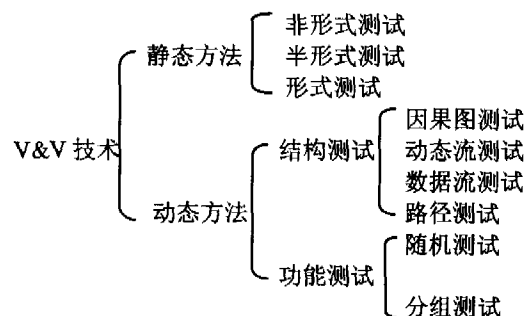


图 2 知识系统 V&V 技术分类

下面我们分别具体介绍知识 V&V 的结构测试和功能测试方法。

结构测试的方法包括很多,主要有:

(1)决策表法:以决策表的方式组织所有的规则,根据规则的前提和结论来检测冗余性和完备性等。决策表是 1974 年由 Montalbano 提出的,用于理顺事物之间的逻辑关系,以

便于检查和分析^[15]。1987年 Cragun 和 Steudel 将决策表用于规则系统知识库的验证。他们通过合理构造知识库中的规则在决策表中的顺序,来检查规则的完备性和一致性(如矛盾规则、冗余规则、遗漏规则等),取得了一些成果^[15]。文[16]提出了基于决策表的知识库局部完备、局部冗余、局部一致和知识库局部完备、局部冗余、局部一致的检查算法。

(2)逻辑的方法:根据知识库构造一些逻辑运算或者逻辑公式,然后进行推导,通过得出的结果对知识库进行验证。Gelfond 和 NscNts 提出逻辑程序设计应该容许经典的否定和作为失败的否定,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,嫌弃肯定性知识^[17]。文[18]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。

(3)Petri 网方法:Petri 网是一种用于系统建模和分析的有用的工具,已经广泛地应用于各个领域的建模与分析^[19,20]。利用 Petri 网为知识库进行建模,找出知识库的冗余性、不一致性、循环性等特征在 Petri 网上的表现形式,然后利用 Petri 网自身的分析技术,如:可达图、关联矩阵、Petri 网语言、Petri 网进程等,来验证知识库的特性。文[21]基于高级 Petri 网系统给出了面向并行工程的知识库建模方法以及知识库异常检测策略。

(4)图论的方法:将知识库作为一个图结构,然后利用图论的相关知识验证知识库的各种特征。文[22]利用图论方法给出非确定推理中规则系统异常检测的策略。

(5)关联矩阵方法:将规则系统表示成一个关联矩阵,利用矩阵乘法和矩阵比较等方法来检测 KB 的异常。

功能测试主要包括:

(1)机器学习法:核心思想是基于机器学习,使用一下学习策略,从给定的知识库中产生一些测试用例,通过这些例子来验证知识库的正确性。

(2)关联方法:将领域知识库中所有的规则所使用的属性

的集合定义为属性空间,将规则视为属性空间上的函数,对知识库的验证就等于规则函数之间的关联关系验证。

(3)求精法:根据情形数据库、统计概念、启发式规则等方法发现知识库中潜在的异常现象,并且对可能的修改方法提供建议。

5 知识系统 V&V 的工具

在知识库建立初期,由于知识库规模较小,内容较简单,规则数目少,用人工方法就可以对知识库运行进护,对其中的事实和规则逐项逐条检查,就可以保证它们的一致性和完整性。但是,随着时间的推移,人们对知识库在最初设计阶段的种种考虑逐渐淡薄,同时由于专家系统在运行过程中又有许多新的知识不断加入知识库,使得知识库的规则越来越多,内容越来越复杂,规模越来越大,知识库中各知识单元之间的相互影响和相互联系就随之变得复杂,难以跟踪和琢磨。在这种情况下,知识库的维护显得更为重要。人工维护方法具有以下弊端:

(1)由于各种客观因素的影响,常会发生输入不完全或输入有错误等现象,这种错误的潜伏性较大,易造成不必要的危害;

(2)由于人的智力和思考范围有限,很难做到知识的一致性,从而造成知识库结构不良,导致系统操作时发生危机;

(3)要求修改知识系统的人员,必须了解这个系统编程;

(4)系统开发周期长,人力消耗大。

基于上述原因,开发一个独立的知识系统维护模块,以便自动检查知识库确保知识的一致性和完整性,已引起专家系统开发者的重视。国际上比较著名的知识验证工具,包括 COVADIS^[23], COVER^[24], CLINT^[25], IN-DEPTH^[26], MELODIA^[27], PREPARE^[28] 等等。表 3 从语言环境、知识表示方法、领域的依赖性、异常检测的类型、知识库的大小等各个方面对比了它们的异同。

表 3 几种著名的知识系统验证工具比较

	COVADIS	COVER	CLINT	IN-DEPTH	MELODIA	PREPARE
开发语言	Lisp	Prolog and C	Prolog	Lisp	Pascal	C
运行环境	Sun Workst.	Sun-4/300	Sun Workst.	Sun-4/260	IBM3090	DEC3100
知识表示	产生式规则	谓词逻辑	谓词逻辑	谓词逻辑	命题逻辑	谓词逻辑
领域依赖性	依赖	独立	依赖	独立	依赖	独立
知识库模型	不分层	不分层	不分层	继承	不分层	不分层
验证方法	逻辑方法	静态检测	机器学习	知识库化简	逻辑	Petri 网模式识别
检测过程领域专家参与	需要	不需要	需要	不需要	不需要	不需要
使用启发式规则	使用	使用	没使用	没使用	没使用	使用
检测的异常类型	一致性	一致性 冗余性 循环性 deficiency	一致性	一致性 冗余性 循环性 无用的知识对象	一致性 冗余性	一致性 冗余性 循环性 完备性
检测效果	只检测	只检测	检测并纠正	只检测	只检测	只检测
知识库规模	200 条规则	550 条规则	不详	334 条规则	10562 条规则	500 条规则

从表 3 可以看出,大多数知识 V&V 工具的知识表示都是基于逻辑的,知识的一致性检测起来比较容易,所有的工具都能对知识的一致性进行检测,知识的冗余性相对于它的性质也较易检测。

6 待解决的问题

随着知识技术的发展和应用,知识系统日益得到重视。决定知识系统性能重要因素之一的知识 V&V 技术引起了足够多的关注,用于知识验证和校验的环境工具也日益增多。但是,在知识的 V&V 方面,还有许多工作值得继续研究:

(1)V&V 的规范:建立知识库异常的语义模型,目前在这方面已经做了一些工作,但还远远不够;同时,有必要建立关于知识异常的标准术语和概念等。

(2)V&V 的阶段:通过前面的分析可以看出:关于知识系统的 V&V 技术,大都集中在系统开发的后一阶段,也就是在编码阶段。需要研究适当的技术和策略,避免知识获取过程中的异常现象,在知识级别上的验证和分析对这个知识系统更重要。因此,在知识获取阶段、知识推理阶段、知识应用阶段等等各个阶段都应该建立和采取知识 V&V 策略。

(3)V&V 的方法:缺乏对知识库异常各个方面均可

V&V 的统一方法;缺乏对多种策略的混合知识表示方法的 V&V 方法。

(4)V&V 的规模:大规模多学科知识的 V&V 需要一套实际的、行之有效的技术。

(5)V&V 的评价:V&V 技术在不同环境下的有效性,对 V&V 方法复杂性的度量,对不同 V&V 策略进行定性和定量的分析比较。

参 考 文 献

- 曹存根,陆跃飞,顾芳,田雯,丰强泽. 大规模知识获取和共享技术. 21 世纪 100 个技术难题,2002
- 陆汝铃主编. 世纪之交的知识工程. 北京:清华大学出版社,2002
- 陆汝铃主编. 知识科学与计算科学. 北京:清华大学出版社,2003
- 张杰. 专家系统若干问题探讨及其在大型聚酸装置中的应用. 上海:华东理工大学档案馆,1995
- IEEE Glossary of Software Eng. Technology. IEEE Standard 610.12-1990,IEEE,1990
- Cao Cungen, Feng Qiangze, Gao Ying, Gu Fang, Si Jinxin, Sui Yuefei, Tian Wen, Wang Haitao, Wang Lili, Zeng Qingtian, Zhang Chunxia, Zheng Yufei, Zhou Xiaobin. Progress in the development of national knowledge infrastructure. Journal of Computer Science and Technology, 2002, 17(5): 523~534
- Cao Cungen. Medical knowledge acquisition from the electronic encyclopedia of China. Lecture Notes in Artificial Intelligence, Beijing, China, 2001
- 陆汝铃,石纯一,张松懋,毛希平,徐晋辉,杨萍,范路. 面向 Agent 的常识知识库. 中国科学(E), 2000, 30(5): 453~463
- 曹存根. 关于建立“中国国家知识基础设施”的建议. 计算机世界报, 1998. 12
- Zeng Qingtian, Cao Cungen, Sui Yuefei. Mathematical Knowledge Acquisition and Knowledge Inheritance Mechanism Based on Ontology. Microelectronics & Computer, 2003, 20(9): 19~27
- 曹存根. 面向专家的知识获取. 科学出版社, 1998
- Agarwal R, Kannan R, Tanniru M. Formal validation of a knowledge-based system using a variation of the Turing Test. Expert Systems With Applications, 1993(6): 181~192
- Kang B H, Gambetta W, Compton, P. Verification and validation with ripple down rules. In: Plant, R. T., Chair, eds. Validation and Verification of Knowledge-Based Systems; 1994 Workshop Program, Seattle, July - August, 1994; Workshop Notes: American Association for Artificial Intelligence
- Kang, Byeong, Gambetta W, Compton P. Verification and validation with ripple down rules. Workshop Notes, Validation and Verification of Knowledge-based Systems, American Association for Artificial Intelligence, 1994
- Cragun B, Steudel H. A Decision Table Based Processor for Checking Completeness and Consistency. International Journal of Man Machine Studies, 1987(26): 633~648
- Vanthienen J, Dries E. Illustration of a decision table tool for specifying and implementing knowledge based systems. In: Proc. 5th Intl. Conf. on Tools with Artificial Intelligence TAI '93, Boston, Mass, Nov. 1993. 198~205
- Kowalski, Sedri F. Logic programs with exceptions. In: Proc. of the 7th Intl. Conf. of Logic Programming. Cambridge, MA: MIT Press, 1990. 598~613
- 王清毅, 陈恩红, 刘贵全, 蔡庆生. 处理知识库中不一致性的超决定逻辑研究. 软件学报, 1998, 9(4): 311~315
- 曾庆田, 吴哲辉. Petri 网的进程网系统. 计算机学报, 2002, 25(12): 1308~1315
- Zeng QT. Behavior descriptions of structure-complex Petri nets based on synchronous composition. Journal of Software, 2004, 15(3): 327~337
- 段华, 曾庆田, 贺国平. 面向并行工程的知识库校验模型与方法研究. 系统工程理论与实践, 2004, 24(3): 79~86
- Valiente. Using layered support graphs for verifying external adequacy in rule-based expert systems. Sigart Bulletin, 1992, 3(1): 20~24
- Rousset M C. On the consistency of Knowledge-bases; the COVADIS system. In: proc. of the ECAI88 Conf. Mancher, Germany, 1988. 79~84
- Precece A, Shinghal R, Batarekh A. Principles and Practice in Verifying Rule-Based Systems. Knowledge Engineering Review, 1992, 7(2): 115~141
- de Raedt L, Sablon G, Bruynooghe. Using Interactive Concept Learning for Knowledge-Base Validation & Verification and test of KBS, Chapter 12, John Wiley & Sons, Chichester, England, 1991. 177~190
- Meseguer P. Incremental Verification of Rule-Based Expert Systems. In: Proc. of the 10th European Conf. on Artificial Intelligence, ECAI-92, Vienna, Austria, 840~844
- Charles E, Dubois O. MELODIA: Logical Methods for Checking K-Bases, In: M. Ayel & J. P. Laurant, eds. Validation, Verification and Test of Knowledge-Based Systems, Chapter 7, Wiley & Sons, Chichester, England, 1991. 95~105
- Zhang D, Nguyen D. PREPARE: A Tool for Knowledge-base Verification. IEEE Transactions on Knowledge and Data Engineering, 1994, 6(6): 983~989

更 正

《计算机科学》2005 年 32 卷第 11 期(P235~P237)刊登王万诚教授“用前馈神经网络对软件理解中函数调用序列的混沌识别”一文中 P237:

(1)取 $M_1 = 100$, 作数据的 200—2200 点的 20 个值, 算出 λ^1 的平均值:

$$\lambda^1 = -0.4818 < 0,$$

M_1 调用序列存在混沌现象。

有误. 特更正如下:

(1)取 $M_1 = 100$, 作数据的 200—2200 点的 20 个值, 算出 λ^1 的平均值:

$$\lambda^1 = -0.4818 < 0,$$

M_1 调用序列不存在混沌现象。

再此, 谨向作者和读者表示歉意。